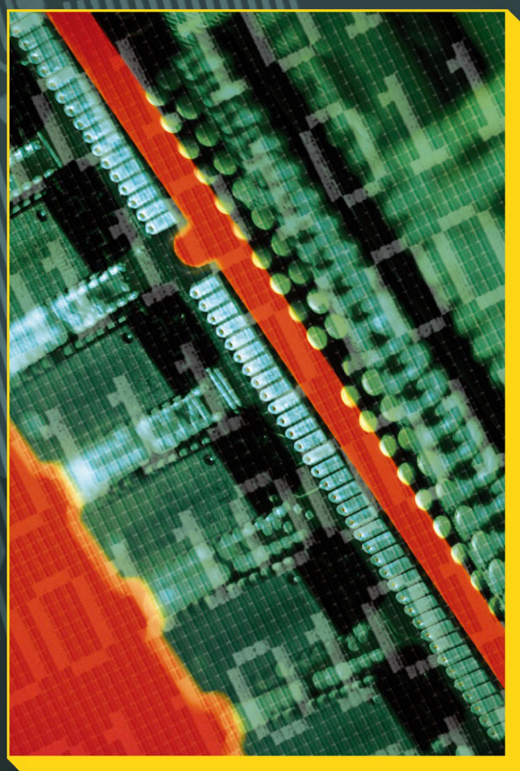
TURING 图灵电子与电气工程丛书

数字电路和数字系统

实验与课程设计实训教程

杨 萍 主编
张旭东 罗 珣 许良凤 编著



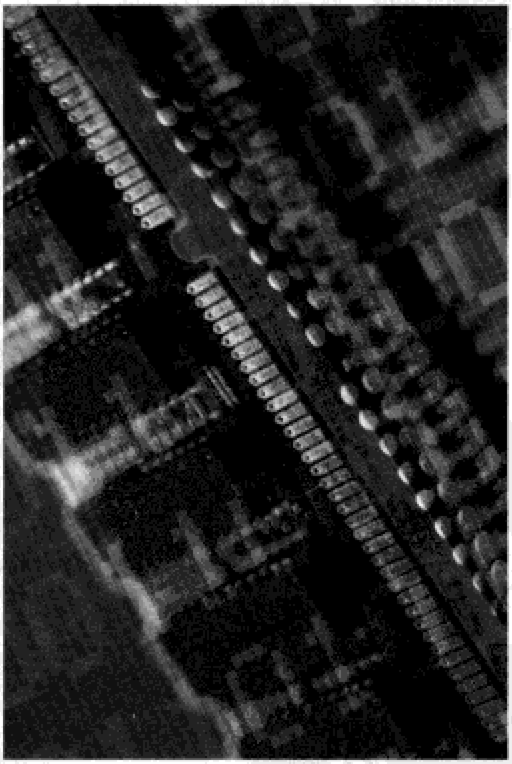
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵电子与电气工程丛书

数字电路和数字系统

实验与课程设计实训教程

杨　萍　主编
张旭东　罗　珣　许良凤　编著



人民邮电出版社
北　京

图书在版编目（CIP）数据

数字电路和数字系统：实验与课程设计实训教程/杨萍主编；张旭东，罗珣，许良凤编著. —北京：人民邮电出版社，2009.5
（图灵电子与电气工程丛书）
ISBN 978-7-115-20460-8

I. 数… II. ①杨… ②张… ③罗… ④许… III. ①数字电路－高等学校－教材 ②数字系统－高等学校－教材 IV. TN79 TP271

中国版本图书馆CIP数据核字（2009）第026784号

内 容 提 要

本书根据国家教育部高等学校电子信息科学与电气信息类基础课程教学指导分委员会提出的“数字电路与逻辑设计”课程教学基本要求，以及可编程逻辑器件（PLD）的实际应用而编写。全书共分为3章，第1章为数字逻辑基础实验，包含20项中小规模数字电路分析与设计的基础实验内容，第2章为数字电路与系统课程设计，介绍了专用集成电路设计和在系统编程技术，第3章主要介绍高密度可编程器件FPGA和模拟可编程器件的开发与应用。

本书贴近实战，内容新颖，可作为高等学校电子信息科学与电气信息类各专业的实践教材，也可供从事电子技术工作的工程技术人员参考。

图灵电子与电气工程丛书

数字电路和数字系统：实验与课程设计实训教程

◆ 主编　杨　萍
编著　张旭东　罗　珣　许良凤
责任编辑　舒　立

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092　1/16
印张：11.75
字数：278 千字　2009 年 5 月第 1 版
印数：1 - 3 000 册　2009 年 5 月河北第 1 次印刷

ISBN 978-7-115-20460-8/TN

定价：25.00元

读者服务热线：（010）88593802　印装质量热线：（010）67129223
反盗版热线：（010）67171154

前　　言

“数字电路”、“数字逻辑”等专业基础课是高等院校电气信息类的主干课程。这些基础课既具有理论性又具有工程实践性，因此，课程实验与课程设计是其中十分重要的实践环节，对培养学生理论联系实际的能力，严谨、科学的实践作风，有着重要的作用。

本书根据国家教育部高等学校电子信息科学与电气信息类基础课程教学指导分委员会2004年8月提出的“数字电路与逻辑设计”课程教学基本要求，根据可编程逻辑器件（PLD）的设计应用，并在总结多年数字电路和数字系统实验与课程设计方面的教学经验基础之上，循序渐进，由浅入深，阐述了数字电路和数字逻辑系统设计的传统方法与基于可编程逻辑器件EDA的现代方法。

在掌握常用仪器的使用和中小规模数字集成电路分析与设计的基础之上，引入专用集成电路设计和在系统编程技术设计训练实验，并结合PLD的实际应用，介绍EDA软件系统开发工具的使用和EDA设计实例等多方面内容。全书共分为3章，第1章为数字逻辑基础实验，共20项，包含中小规模数字电路分析与设计的基础实验内容；第2章为数字电路与系统课程设计，介绍了专用集成电路设计和在系统编程技术，设置了8项小的数字系统设计训练实验，列出26项课程设计参考选题；第3章主要介绍高密度可编程器件FPGA和模拟可编程器件的开发与应用，编制了9项基于FPGA的应用实验，列出15项可编程器件课程设计综合选题。附录汇编了常用数字集成电路名称、型号及引脚排列图，可供学生在实验、课程设计和毕业设计过程中查阅参考。

本书作为平台课程的教材，实验与课程设计适应面广，针对性强。对于电气、计算机类专业学生，选用第1章、第2章作为基本内容，第3章作为选修或选读内容。对于通信、电子类专业学生，则3章内容都要学会、贯通，才能达到教学基本要求。本教材一般使用两个学期，便于课程与课程设计内容的衔接，全面提高教学质量。

全书包含37项数字电路与系统的实验，41个课程设计综合选题，实训效果好，特别适合作为实验单独设课的教材。

本书由杨萍担任主编，第1章由杨萍和许良凤编写，第2章由杨萍和罗珣编写，第3章由张旭东编写，杨萍负责全书统稿。梁华国教授和徐小红博士对初稿进行了认真细致的审阅，提出了许多宝贵的意见和修改建议，在此向他们表示衷心的感谢。

在本教材的编写过程中，参考了多位学者和专家的著作，在此表示诚挚的谢意。

由于编者的水平有限，书中难免有疏漏和不当之处，恳请各位读者提出批评与改进意见。

编　者

2008年8月

目　　录

第1章　数字逻辑电路基础实验

1.1　数字逻辑电路实验基本知识

1.1.1　数字集成电路封装

中小规模数字IC中最常用的是TTL电路和CMOS电路。TTL器件型号以74（或54）作为前缀，称为74/54系列，如74LS10、74F181、54S86等。中小规模CMOS数字集成电路主要是4XXX/45XX（X代表0～9的数字）系列、高速CMOS电路HC（74HC）系列、与TTL兼容的高速CMOS电路HCT（74HCT）系列。TTL电路与CMOS电路各有优缺点，TTL速度快，CMOS功耗小、电源范围大、抗干扰能力强。由于TTL在世界范围内应用极广，因此在数字电路教学实验中，我们主要使用TTL74系列电路作为实验用器件，采用单一+5V作为供电电源。

数字IC器件有多种封装形式。为了教学实验方便，实验中所用的74系列器件采用双列直插式封装。图1-1是双列直插封装的正面示意图。双列直插封装有以下特点。



图1-1　双列直插式封装图

（1）从正面（上面）看，器件一端有一个半圆的缺口，这是正方向的标志。缺口左边的引脚为1号引脚，引脚号按逆时针方向增加。图1-1中的数字表示引脚号。双列直插封装IC的引脚数有14、16、20、24、28等若干种。

（2）双列直插器件有两列引脚。引脚之间的间距是2.54mm。两列引脚之间的距离有宽（15.24mm）和窄（7.62mm）两种，该距离能够稍做改变，但引脚间距不能改变。将器件插入实验台上的插座或者从插座拔出时均要小心，不要弯曲或折断器件引脚。

（3）74系列器件一般左下角的最后一个引脚是GND，右上角的引脚是Vcc。例如，14引脚器件中，引脚7是GND，引脚14是Vcc；20引脚器件中，引脚10是GND，引脚20是Vcc。但也有一些例外，例如16引脚的双JK触发器74LS76中，引脚13（不是引脚8）是GND，引脚5（不是引脚16）是Vcc。所以使用集成电路器件时要先看清它的引脚图，找对电源和地，避免因接线错误造成器件损坏。

数字电路综合实验中，使用的复杂可编程逻辑器件MACH4—64/32（或者ISP1016）是44引脚的PLCC（Plastic Leaded Chip Carrier）封装，图1-2是封装正面图。器件上的小圆圈指示引脚1，引脚号按逆时针方向增加，引脚2在引脚1的左边，引脚44在引脚1的右边。MACH4—64/32电源引脚号、地引脚号与ISP1016的不同，千万不要插错PLCC插座。插PLCC器件时，器件的左上角（缺角）要对准插座的左上角。拔PLCC器件应使用专门的起拔器。

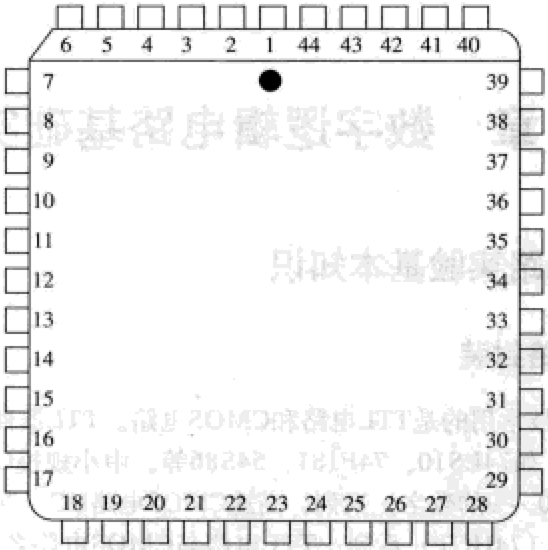


图1-2　PLCC封装图

TDS实验台上的接线采用自锁紧插头和插孔（插座）。使用自锁紧插头和插孔接线时，首先把插头插进插孔中，然后将插头按顺时针方向轻轻一拧则锁紧。拔出插头时，首先按逆时针方向轻轻拧一下插头，使插头和插孔之间松开，然后将插头从插孔中拔出。不要使劲拔插头，以免损坏插头和连线。

注意：不能带电插、拔器件。插、拔器件只能在关断+5V电源的情况下进行。

1.1.2　数字电路测试及故障查找和排除

设计好一个数字电路后，要对其进行测试，以验证设计是否正确。测试过程中，发现问题要分析原因，找出故障所在，并解决它。数字电路实验也遵循这些原则。

1. 数字电路测试

数字电路测试大体上分为静态测试和动态测试两部分。静态测试指的是，给定数字电路若干组静态输入值，测试数字电路的输出值是否正确。数字电路设计好后，在实验台上连接成一个完整的线路。线路的输入接电平开关输出，线路的输出接电平显示单元（指示灯），按功能表或状态表的要求，改变输入状态，观察输入和输出之间的关系是否符合设计要求。静态测试是检查设计是否正确、接线是否无误的重要一步。

在静态测试基础上，按设计要求在输入端加动态脉冲信号，观察输出端波形是否符合设计要求，这就是动态测试。有些数字电路只需进行静态测试，而有些数字电路则必须进行动态测试。一般而言，时序电路应进行动态测试。

2. 数字电路的故障查找和排除

在数字电路实验中，出现问题是难免的。重要的是分析问题，找出问题的原因，从而解决它。一般而言，有四个方面的原因产生问题（故障）：器件故障、接线错误、设计错误和测试方法不正确。在查找故障过程中，首先要了解以上这些经常发生的典型故障

原因。

(1) 器件故障

器件故障是器件失效或器件接插问题引起的故障，表现为器件工作不正常。不言而喻，器件失效肯定会引起工作不正常，这需要更换一个好器件。器件接插问题［如引脚折断或者器件的某个（或某些）引脚没插到插座中等］也会使器件工作不正常。器件接插问题有时不易发现，需仔细检查。判断器件失效的方法是用集成电路测试仪测试器件。需要指出的是，一般的集成电路测试仪只能检测器件的某些静态特性。对负载能力等静态特性和上升沿、下降沿、延迟时间等动态特性，一般的集成电路测试仪不能测试，必须使用专门的集成电路测试仪。

(2) 接线错误

接线错误是最常见的错误。据统计，在教学实验中，大约70%以上的故障是由接线错误引起的。常见的接线错误包括忘记接器件的电源和地；连线与插孔接触不良；连线经多次使用后，外面塑料包皮有可能完好，但内部线断；连线多接、漏接、错接；连线过长、过乱造成干扰等。接线错误造成的现象多种多样，例如器件的某个功能块不工作或工作不正常，器件不工作或发热，电路中一部分工作状态不稳定等。解决这些问题首先基于熟悉所用器件的功能及其引脚号，掌握器件每个引脚的功能；其次在实验中，器件的电源和地一定要接对和接好；检查连线和插孔接触是否良好；检查连线有无错接、多接、漏接；检查连线中有无断线。最重要的是，接线前要画出接线图，按图接线，不要凭记忆随想随接，接线要规范、整齐，尽量走直线、短线，以免引起干扰。

(3) 设计错误

设计错误自然会造成与预想的结果不一致，究其原因就是对实验要求没有吃透，或者是对所用器件的原理没有掌握。因此实验前一定要理解实验要求，掌握实验线路原理，精心设计。初始设计完成后一般应对设计进行优化，最后画好逻辑图和接线图。

(4) 测试方法不正确

如果不发生前面所述三种错误，实验一般会成功。但有时测试方法不正确也会引起观测错误。例如，对于一个稳定的波形，如果用示波器观测，而示波器没有同步，就会造成波形不稳的假象。因此要学会正确使用所用仪器和仪表。在数字电路实验中，尤其要学会正确使用示波器。在测试数字电路时，由于测试仪器仪表要加到被测电路上，这对被测电路来说相当于一个负载，因此也有可能引起电路本身工作状态的改变，这点应引起足够注意。不过，在数字电路实验中，这种现象很少发生。

当实验中发现结果与预期不一致时，千万不要慌乱。应仔细观测现象，冷静思考问题所在。首先检查仪器仪表的使用是否正确。在正确使用仪器仪表的前提下，按逻辑图和接线图逐级查找问题所在。通常从出现问题的地方，一级一级向前测试，直到找出故障的初始发生位置。在故障的初始位置处，检查连线是否正确。前面已说过，实验故障绝大部分是由接线错误引起的，因此检查一定要认真仔细。确认接线无误后，检查器件引脚是否全部正确插进插座中，有无引脚折断、弯曲和错插问题。确认无上述问题后，取下器件测试，以检查器件好坏，或者直接换一个好器件。如果器件和接线都正确，则需考虑设计问题。

1.2　数字逻辑电路基础实验

实验一　晶体管开关特性及其应用

一、实验目的

(1) 掌握晶体二极管和晶体三极管的开关特性。

(2) 掌握限幅器、钳位器和反相器的基本工作原理。

(3) 掌握由晶体管实现的基本逻辑门电路。

二、实验原理

1. 晶体二极管的开关特性

由于晶体二极管具有单向导电性，故其开关特性表现在正向导通与反向截止两种不同状态间的转换。如图1-3所示电路，输入端施加方波激励信号V_i，由于二极管电容的存在，因而有充电、放电和存储电荷的建立与消散的过程。因此，当加在二极管上的电压突然由正向偏置（$+V_1$）变为反向偏置（$-V_2$）时，二极管并不立即截止，而是出现一个较大的反向电流I_R（$I_R=-V_2/R$），并维持一段时间t_s（称为存储时间）后，电流才开始减小，再经过t_t（称为渡越时间）后，反向电流才等于静态特性上的反向电流$0.1I_R$。$t_{re}=t_s+t_t$叫作反向恢复时间。t_{re}与二极管的结构有关，PN结面积小，电容小，存储电荷就少，t_s就短，同时也与正向导通电

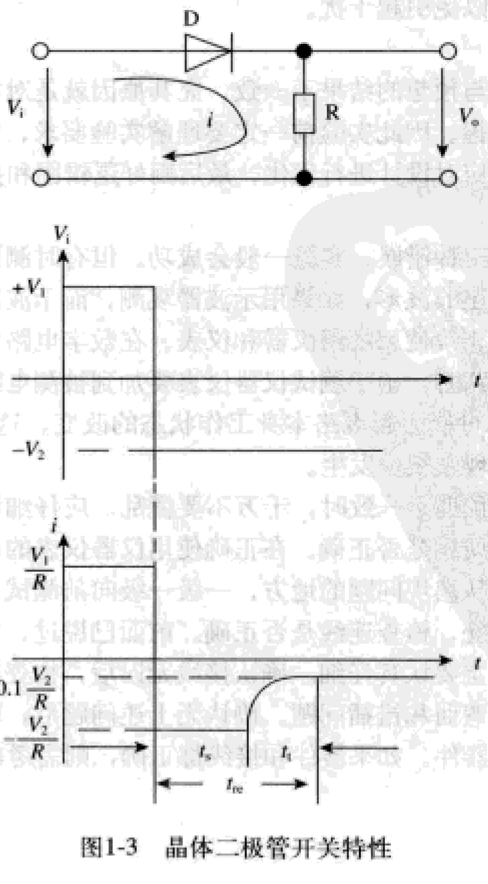


图1-3　晶体二极管开关特性

流和反向电流有关。当选定二极管后，减小正向导通电流和增大反向驱动电流，可加速电路的转换过程。

2. 晶体三极管的开关特性

晶体三极管的开关特性是指它从截止到饱和、饱和到截止的状态转换过程，而且这种转换需要一定的时间才能够完成。图1-4为晶体三极管开关特性电路图与对应的输入输出波形图。电路的输入端施加一个足够幅度（在$-V_2$和$+V_1$之间变化）的矩形脉冲电压V_i激励信号，则晶体管T的集电极输出电流i_c和输出电压V_o的波形已不是和输入波形一样的理想方波，其起始部分和平顶部分都延迟了一段时间，上升沿和下降沿都变得平缓了。为了对晶体三极管开关特性进行定量描述，通常引入以下几个参数来表征。

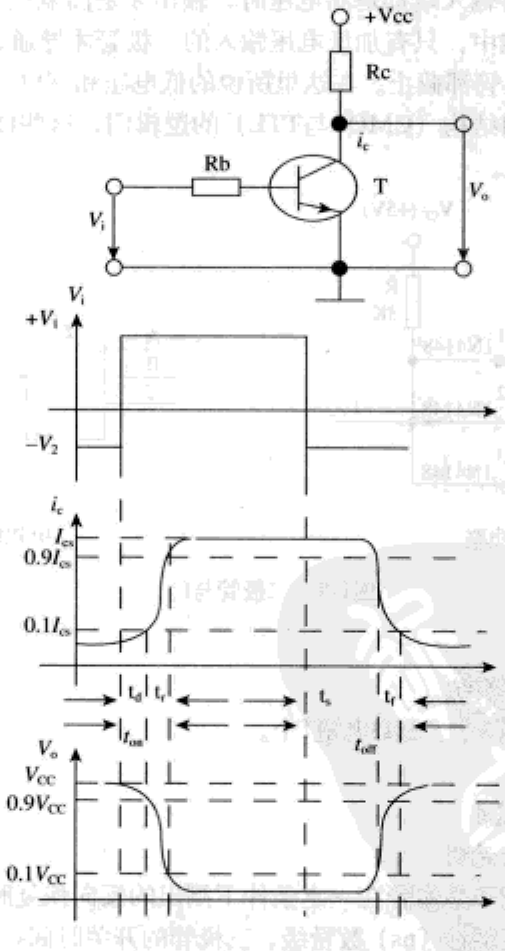


图1-4 晶体三极管开关特性

延迟时间t_d —— 从$+V_1$加入，集电极电流i_c上升到$0.1I_{cs}$所需的时间；

上升时间t_r —— i_c从$0.1I_{cs}$增长到$0.9I_{cs}$所需的时间；

存储时间t_s —— 从$-V_2$加入，集电极电流i_c下降到$0.9I_{cs}$所需的时间；

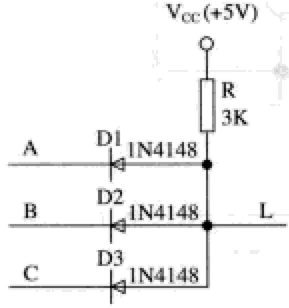
下降时间t_f —— i_c从$0.9I_{cs}$增长到$0.1I_{cs}$所需的时间；

以上参数称为三极管的开关时间参数，它们都是以集电极电流i_c的变化为基准的。通常把t_{on}

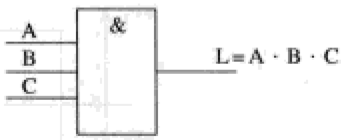
$=t_d+t_r$称为导通时间，它反映了三极管从截止到饱和所需的时间；而把$t_{off}=t_s+t_f$称为关闭时间，它反映了三极管从饱和到截止所需的时间。导通时间和关闭时间总称为三极管的开关时间，它随三极管类型不同而有很大差别，一般在几十到几百纳秒的范围内。

3. 利用二极管与三极管的非线性特性构成限幅器和钳位器，实现基本逻辑电路

二极管限幅器是利用二极管导通时和截止时呈现的阻抗不同来实现限幅，其限幅电平由外接偏压决定。三极管则利用其截止和饱和特性实现限幅。钳位的目的是将脉冲波形的顶部或底部钳制在一定的电平上。正是上面这些特性实现了基本数字逻辑门电路。相关基本逻辑门电路的实现可参考相关教材，在此仅举与门电路说明。图1-5a表示由二极管组成的与门电路，图1-5b为它的逻辑符号。图中A、B、C为输入端，L为输出端。输入信号为+5V或0V。与逻辑的要求是，只有所有输入端都是高电压时，输出才是高电压，否则输出就是低电压。由此可见，与门几个输入端中，只有加低电压输入的二极管才导通，并把输出L钳制在低电压，而加高电压输入的二极管都截止。在这里所说的低电压和高电压不是一个固定的值，而是一个电压范围，对于不同结构（CMOS与TTL）的逻辑门，这些技术参数都不一样。这将在下面实验中进行详细阐述。



(a) 电路



(b) 逻辑符号

图1-5 二极管与门

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）1N4148开关二极管3个、3kΩ电阻1个。

（3）数字万用表。

四、实验内容及实验步骤

1. 晶体管开关特性实验说明

器件手册中一般都给出了晶体管在一定条件下测出的反向恢复时间和开关时间。开关二极管的反向恢复时间通常在纳秒（ns）数量级，三极管的开关时间在几十到几百纳秒的范围。从纳秒数量级的量纲分析可知，测量晶体管开关特性时对仪器要求较高，普通的双踪示波器无法观察和测量相关参数，故不要求做本实验，使用晶体管时查找相关器件手册即可。若有精密仪器，有兴趣的同学不妨参考图1-3与图1-4中的电路图来设计实验电路，并测量并记录相关参数。

2. 基本逻辑门电路实验

在实验箱元件库模块中，按图1-5所示连接实验电路，A、B、C作为输入，地表示低电

压，+5V表示高电平，根据A、B、C共8种输入情况，用数字万用表测量出输出量L的值，并列表记录之。

五、实验预习要求

（1）复习晶体管开关特性原理。

（2）熟悉基本逻辑门的组成原理。

（3）熟悉实验箱基本使用方法和使用技巧。

六、实验报告要求

（1）总结晶体管开关特性。

（2）画出二极管或门、三极管反相器的电路图，分析电路实现或门和反相器的原理，并写出相应真值表。

实验二 集成逻辑门电路参数测试

一、实验目的

（1）掌握TTL集成与非门的主要性能参数及测试方法。

（2）掌握TTL器件的使用规则。

（3）熟悉数字电路测试中常用电子仪器的使用方法。

二、实验原理

制造TTL门电路的厂家，通常都要为用户提供各种逻辑器件的数据手册。本实验采用二输入四与非门74LS00（它的俯视图见附录）来学习TTL各项技术参数。74LS00内含有四个相互独立的与非门，每个与非门有两个输入端。

1. TTL集成与非门的逻辑功能

单个与非门的逻辑功能框图如图1-6所示，当输入端中有一个或一个以上是低电平时，输出为高电平；只有输入端输入全都为高电平时，输出端才是低电平。

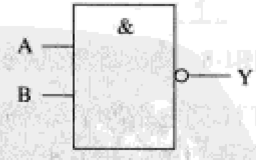


图1-6 74LS00的逻辑图

2. TTL集成与非门的主要参数

主要参数有输出高电平V_{OH}、输出低电平V_{OL}、输入短路电流I_{is}、扇出系数N_0、电压传输特性和平均传输延迟时间t_{pd}等。

（1）TTL门电路的输出高电平V_{OH}

V_{OH}是与非门有一个或多个输入端接地或接低电平时的输出电压值，此时与非工作管处于截止状态。空载时，V_{OH}的典型值为3.6V，接有拉电流负载时，V_{OH}下降。

（2）TTL门电路的输出低电平V_{OL}

V_{OL}是与非门所有输入端都接高电平时的输出电压值，此时与非门处于饱和导通状态。空载时，它的典型值约为0.2V，接有灌电流负载时，V_{OL}将上升。

（3）TTL门电路的输入短路电流I_{is}

它是指当被测输入端接地、其余端悬空、输出端空载时，由被测输入端输出的电流值，测试电路图如图1-7所示。

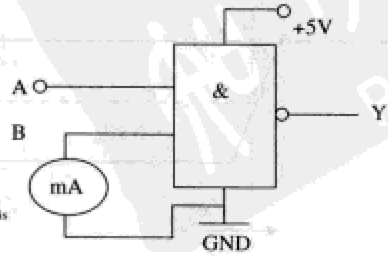


图1-7 I_{is}的测试电路图

(4) TTL门电路的扇出系数N_0

扇出系数N_0指门电路能驱动同类门的个数，它是衡量门电路负载能力的一个参数。TTL集成与非门有两种不同性质的负载，即灌电流负载和拉电流负载。因此，它有两种扇出系数，即低电平扇出系数N_{0L}和高电平扇出系数N_{0H}。通常有$I_{iH}<I_{iL}$，则$N_{0H}>N_{0L}$，故常以N_{0L}作为门的扇出系数。

N_{0L}的测试电路如图1-8所示。门的输入端全部悬空，输出端接灌电流负载R_L，调节R_L使I_{OL}增大，V_{OL}随之增高，当V_{OL}达到V_{OLm}（手册中规定低电平规范值为0.4V）时的I_{OL}就是允许灌入的最大负载电流，则

$$N_{0L}=I_{OL}\div I_{is}，通常N_{0L}>8$$

(5) TTL门电路的电压传输特性

门的输出电压V_o随输入电压V_i而变化的曲线$V_o=f(V_i)$称为门的电压传输特性，通过它可读得门电路的一些重要参数，如输出高电平V_{OH}、输出低电平V_{OL}、关门电平V_{off}、开门电平V_{ON}等值。测试电路如图1-9所示，采用逐点测试法，即调节R_w，逐点测得V_i及V_o，然后绘成曲线。

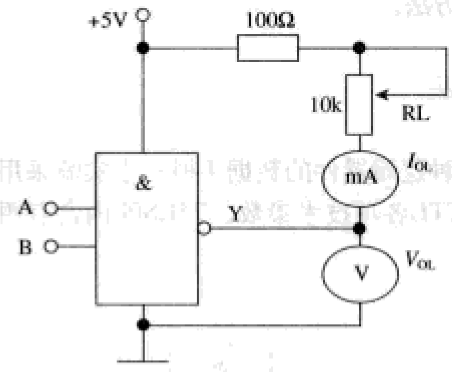


图1-8 扇出系数测试电路

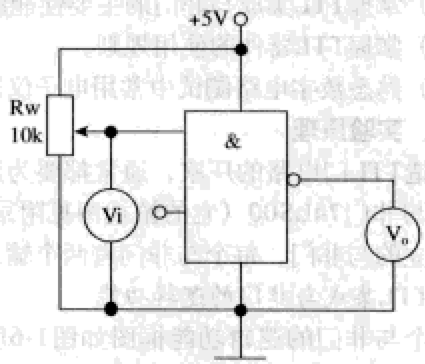
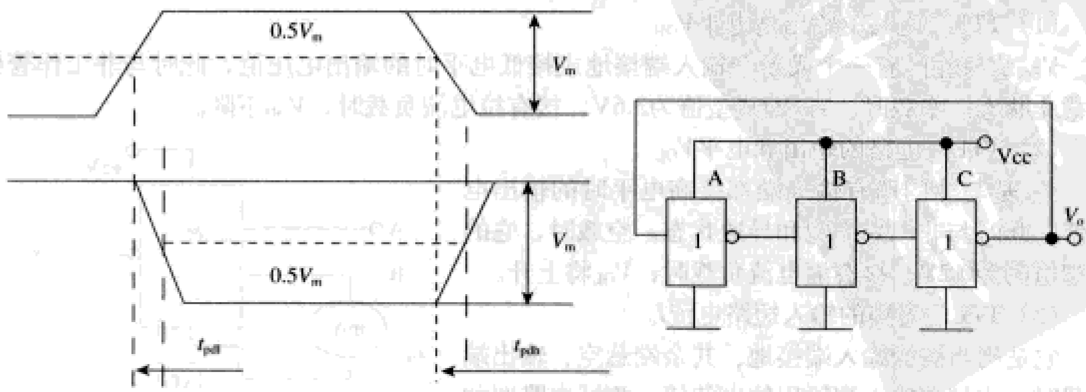


图1-9 电压传输特性测试电路

(6) TTL门电路的平均传输延迟时间t_{pd}

t_{pd}是衡量门电路开关速度的参数，它意味着门电路在输入脉冲波形的作用下，其输出波形相对于输入波形延迟了多少时间。具体地说，是指输出波形边沿的$0.5V_m$至输入波形对应边沿$0.5V_m$点的时间间隔，如图1-10所示。一般传输延迟时间短，为纳秒数量级。



(a) 传输延迟特性 (b) t_{pd}的测试电路

图1-10 TTL门电路的平均传输延迟时间

图1-10a中的t_{pdL}为导通延迟时间，t_{pdH}为截止延迟时间，平均传输时间为

$$t_{pd}=（t_{pdL}+t_{pdH}）/2$$

t_{pd}的测试电路如图1-10b所示，由于门电路的延迟时间较小，直接测量时对信号发生器和示波器的性能要求较高，故实验采用测量由奇数个非门组成的环形振荡器的振荡周期T来求得。其工作原理是：假设电路在接通电源后某一瞬间，电路中的A点为逻辑“1”，经过三级门的延时后，使A点由原来的逻辑“1”变为逻辑“0”；再经过三级门的延时后，A点重新回到逻辑“1”。电路的其他各点电平也随着变化。说明使A点发生一个周期的振荡，必须经过6级门（两次循环）的延迟时间。因此平均传输延迟时间为$t_{pd}=T/6$。TTL电路的t_{pd}一般为10ns~40ns。

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）二输入端四与非门74LS00（1片）、六反相器 74LS04（1片）、100Ω电阻（1个）。

（3）数字万用表。

（4）双踪示波器。

四、实验内容及实验步骤

（1）在实验箱IC插座模块找一个DIP14的插座插上芯片74LS00，并在DIP14插座的第7脚接上实验箱的地（GND），第14脚接上电源+5V（V_{CC}）。其他引脚的连线参考具体的线路图，测试与非门的逻辑功能，记录之。

（2）按照实验原理第二部分说明，用万用表测出TTL与非门电路的V_{OH}和V_{OL}值。

（3）按图1-7所示连线，用万用表的电流档测出TTL与非门电路的输入短路电流I_{is}。

（4）按图1-8所示连线，先用万用表测试使V_{OL}等于V_{OLm}（0.4V），再把万用表调到电流档串入电路测出I_{OL}，求得扇出系数N_0。

（5）按图1-9所示连线，调节电位器R_W，使V_i从0V向高电平变化，逐点测量V_i和V_o,将结果记入下表中。

V_i	0	0.2	0.4	0.6	0.8	1.0	1.5	2.0	2.5	3.0	3.5	4.0	…
V_o													

（6）按图1-10所示连线，V_{CC}接+5V，记录V_o的波形，测量出V_o的周期T，并算出t_{pd}。

五、实验预习要求

（1）复习TTL门电路的工作原理。

（2）熟悉实验所用集成门电路引脚功能。

（3）画出实验内容中的测试电路与数据记录表格。

六、实验报告要求

（1）记录整理实验结果，并对结果进行分析。

（2）画出实测的电压传输特性曲线，并从中读出各有关参数值。

（3）思考TTL门电路的无用输入端是否能悬空或接高电平？为什么？

七、TTL门电路的使用规则小结

（1）接插集成块时，要认清定位标记，不能插反。器件必须先接上电源，再接上信号源。

(2) 对电源要求比较严格，只允许在5V±10%的范围内工作，电源极性不可接错。

(3) 普通TTL与非门不能并联使用（集电极开路门与三态输出门电路除外），否则不仅会使电路逻辑功能混乱，还会导致器件损坏。

(4) 必须正确处理闲置输入端。闲置输入端处理方法如下。

① 悬空，相当于正逻辑"1"，对于一般小规模集成电路的数据输入端，实验时允许悬空处理。但这样易受外界干扰，导致电路的逻辑功能不正常。因此，对于接有长线的输入端、中规模以上的集成电路和使用集成电路较多的复杂电路，所有的控制输入端必须按逻辑要求接入电路，不允许悬空。

② 直接接电源电压V_{CC}（也可串入一只1kΩ~10kΩ的固定电阻）或接至某一固定电压（+2.4V<V<4.5V）的电源上，或与输入端为接地的多余与非门的输出端相接。

③ 若前级驱动能力允许，可以与使用的输入端并联。

(5) 负载个数不能超过允许值。

(6) 输出端不允许直接接地或直接接+5V电源，否则会损坏器件。有时为了使后极电路获得较高的输出电平，允许输出端通过电阻接至V_{CC}，一般取电阻值为3kΩ~5.1kΩ。

实验三 基本逻辑门电路的功能测试

一、实验目的

(1) 测试TTL集成芯片中的与门、或门、非门、与非门、或非门与异或门的逻辑功能。

(2) 了解测试的方法与测试的原理。

二、实验原理

实验中用到的基本门电路的符号如图1-11所示。

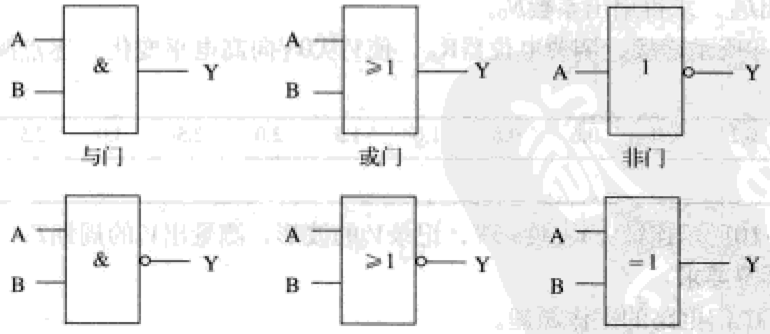


图1-11 基本门电路的逻辑符号

在测试芯片逻辑功能时，输入端用逻辑电平输出（开关）单元输入高低电平，然后使用逻辑电平显示（指示灯）单元显示输出的逻辑功能。

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 芯片与门74LS08、或门74LS32、非门74LS04、与非门74LS00、或非门74LS02（或74LS28）、异或门74LS86各一片。

四、实验内容及实验步骤

(1) 依次选用芯片74LS08、74LS32、74LS04、74LS00、74LS02（或74LS28）、74LS86做实验，在实验箱IC插座模块找到相应引脚数目的IC插座，插上并保持连接正常。

(2) 对照附录的相应芯片引脚图，按照芯片的引脚分布图接线，注意确保电源V_{CC}（+5V）输入脚和地输入脚的连接，芯片输入端连接到逻辑电平输出（开关）单元，通过逻辑电平输出单元控制输入电平。芯片输出端连接到逻辑电平显示（指示灯）单元，输出高电平时对应的发光二极管亮，否则不亮。

(3) 按照芯片各逻辑门的真值表检验芯片的逻辑功能芯。

五、实验预习要求

(1) 掌握集成芯片74LS08、74LS32、74LS04、74LS00、74LS02（或74LS28）、74LS86的引脚分布图。

(2) 分别列出芯片74LS08、74LS32、74LS04、74LS00、74LS02（或74LS28）、74LS86的真值表。

六、实验报告要求

(1) 画好实验中各门电路的真值表表格，将实验结果填写到表中。

(2) 根据实验结果，写出各逻辑门的逻辑表达式，并判断逻辑门的好坏。

实验四 集电极开路门和三态输出门

一、实验目的

(1) 掌握TTL集电极开路门（OC门）的逻辑功能及应用。

(2) 了解集电极负载电阻R_L对集电极开路门的影响。

(3) 掌握TTL三态输出门（3S门）的逻辑功能及应用。

二、实验原理

在数字系统中，有时需要把两个或两个以上集成逻辑门的输出端直接并接在一起，完成一定的逻辑功能。对于普通的TTL电路，由于输出级采用了推拉式输出电路，无论输出是高电平还是低电平，输出阻抗都很低，因此通常不允许将它们的输出端并接在一起使用。而集电极开路门和三态输出门是两种特殊的TTL门电路，它们允许把输出端直接并接在一起使用，也就是说，它们都具有“线与”的功能。

1. TTL集电极开路门（OC门）

本实验所用OC门型号为二输入四与非门74LS03，引脚排列见附录1。工作时，输出端必须通过一只外接电阻R_L和电源Ec相连接，以保证输出电平符合电路要求。

OC门的应用主要有下述三个方面。

(1) 凭借电路的“线与”特性，可以完成某些特定的逻辑功能。如图1-12所示，将两个OC门输出端直接并接在一起，则它们的输出为

$$F = F_A \cdot F_B = \overline{A_1A_2} \cdot \overline{B_1B_2} = \overline{A_1A_2 + B_1B_2}$$

即把两个（或两个以上）OC与非门“线与”可完成“与或非”的逻辑功能。

(2) 实现多路信息采集，使两路以上的信息共用一个传输通道（总线）。

(3) 实现逻辑电平转换，以推动荧光数码管、继电器和MOS器件等多种器件。

OC门输出并联运用时负载电阻R_L的选择如下。

如图1-13所示，电路由n个OC与非门"线与"驱动有m个输入端的N个TTL与非门，为保证OC门输出电平符合逻辑要求，负载电阻R_L阻值的选择范围为

$$R_{L\max}=\frac{E_C-V_{OH}}{nI_{OH}+mI_{iH}},\quad R_{L\min}=\frac{E_C-V_{OL}}{I_{LM}-NI_{iL}}$$

式中：I_{OH}——OC门输出管截止时（输出高电平V_{OH}）的漏电流（约为50μA）

I_{LM}——OC门输出低电平V_{OL}时允许最大灌入负载电流（约为20mA）

I_{iH}——负载门高电平输入电流（<50μA）

I_{iL}——负载门低电平输入电流（<1.6mA）

E_C——R_L外接电源电压

n——OC门个数

N——负载门个数

m——接入电路的负载门输入端总个数

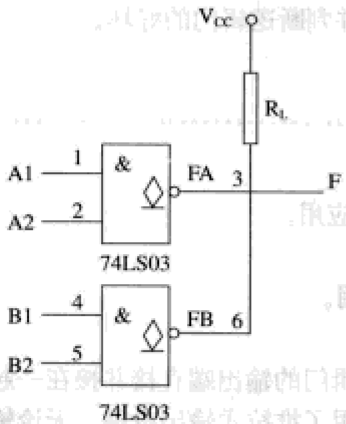


图1-12 OC与非门"线与"电路

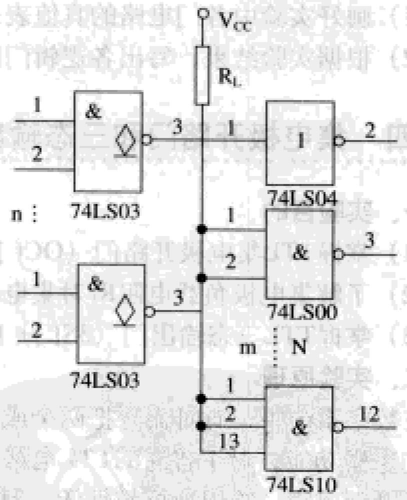


图1-13 OC与非门负载电阻R_L的确定

R_L值必须小于$R_{L\max}$，否则V_{OH}将下降；R_L值必须大于$R_{L\min}$，否则V_{OL}将上升。又由于R_L的大小会影响输出波形的边沿时间，在工作速度较高时，R_L应尽量选取得接近$R_{L\min}$。

2. TTL三态输出门（3S门）

TTL三态输出门是一种特殊的门电路，它与普通的TTL门电路结构不同，它的输出端除了通常的高电平、低电平两种状态外（这两种状态均为低阻状态），还有第三种输出状态——高阻态。处于高阻态时，电路与负载之间相当于开路。三态输出门按逻辑功能及控制方式来分有各种不同类型，本实验所用三态门的型号是74LS125三态输出四总线缓冲器，图1-14是三态输出四总线缓冲器的逻辑符号，它有一个控制端（又称为禁止端或使能端）$\overline{E}$，$\overline{E}=0$为正常

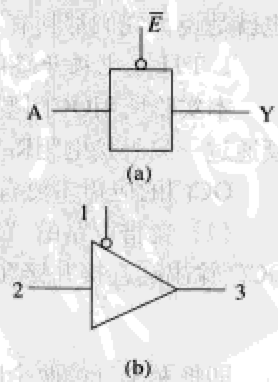


图1-14 三态输出四总线缓冲器的逻辑符号

工作状态，实现$Y=A$的逻辑功能；$\bar{E}=1$为禁止状态，输出Y是高阻态。这种在控制端加低电平电路才能正常工作的方式称低电平使能。74LS125的引脚排列见图1-15。

表1-1 74LS125功能表

输入		输出
$\bar{E}$	A	F
0	0	0
	1	1
1	0	高阻
	1	

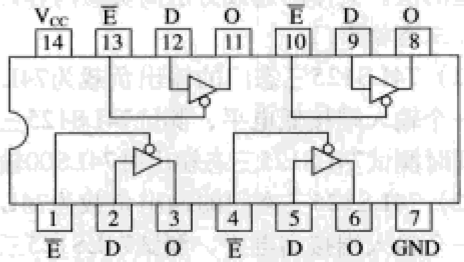


图1-15 74LS125引脚排列图

三态电路的主要用途之一是实现总线传输，即用一个传输通道（称总线），以选通方式传送多路信息。使用时，要求只有需要传输信息的三态控制端处于使能态（$\bar{E}=0$）其余各门皆处于禁止状态（$\bar{E}=1$）。由于三态门输出电路结构与普通TTL电路相同，显然，若同时有两个或两个以上三态门的控制端处于使能态，将出现与普通TTL门“线与”运用时同样的问题，因而是绝对不允许的。

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 芯片74LS00、74LS03、74LS10、74LS04和74LS125各一片。

(3) 200Ω电阻和3MΩ~5MΩ电阻（由3.3MΩ、680kΩ等组成）。

(4) 数字万用表和双踪示波器。

四、实验内容及实验步骤

1. TTL集电极开路与非门74LS03负载电阻R_L的确定

在数字逻辑电路实验箱的IC插座模块中找一个DIP14的插座插上芯片74LS03，并在DIP14插座的第7脚接上实验箱的地（GND），第14脚接上电源+5V（V_{CC}），输入引脚接逻辑电平输出拨位开关。芯片的引脚分配请参考附录1或其他资料。

用两个集电极开路与非门“线与”来驱动一个TTL非门，按图1-16连接实验电路。负载电阻R_L由一个200Ω电阻和一个10kΩ电位器R_W串接而成，取E_C=5V。接通电源，用逻辑开关改变两个OC门的输入状态，先使OC门“线与”输出低电平，调节R_W使V_{OL}=0.3V，测得此时的R_L即为R_{Lmin}，再使电路输出高电平，需要重新调节负载R_L的组成（大概R_L在3MΩ~5MΩ情况下才能使V_{OH}=3.5V），重新调节好负载R_L使V_{OH}=3.5V，测得此时的R_L即为R_{Lmax}。

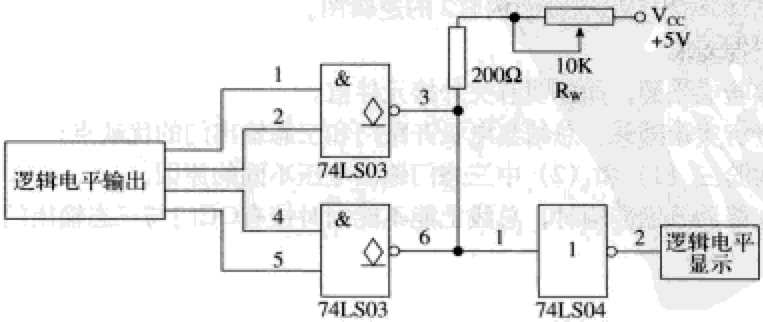


图1-16 74LS03负载电阻的确定

2. 集电极开路门的应用

用OC门实现 $F = A\overline{B} + CD + E\overline{F}$，实验时输入变量只允许用原变量，外接负载电阻$R_L$自取合适的值。具体的连线方法同实验内容1。

3. 三态输出门

（1）74LS125三态门的输出负载为74LS00的一个与非门输入端。74LS00同一个与非门的另一个输入端接低电平，测试74LS125三态门三态输出、高电平输出和低电平输出的电压值。同时测试74LS125三态输出时74LS00输出值。

（2）74LS125三态门的输出负载为74LS00的一个与非门输入端。74LS00同一个与非门的另一个输入端接高电平，测试74LS125三态门三态输出、高电平输出、低电平输出的电压值。同时测试74LS125三态输出时74LS00输出值。

（3）用74LS125实现总线传输

将四个三态缓冲器按图1-17接线，输入端按图加输入信号，控制端接逻辑开关，输出端接LED或示波器，先使4个三态门的控制端均为高电平"1"，即处于禁止状态。然后依次使一个门选通，观察LED或示波器的显示情况。注意：不能让两个门同时选通，图1-17中四个选通信号只能选通一个有效。记录实验结果。

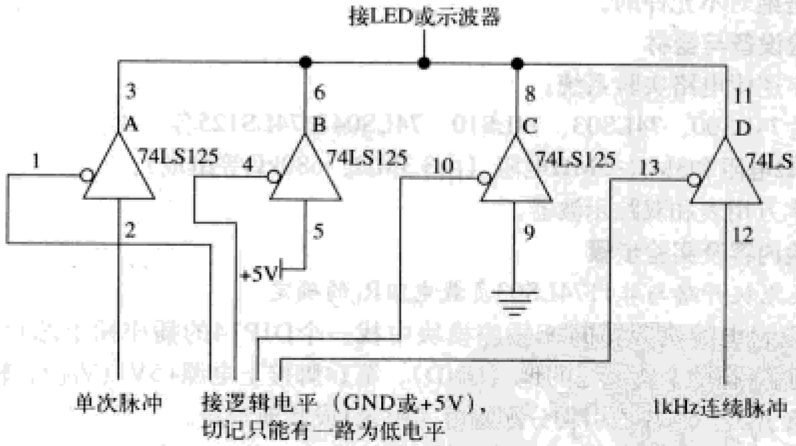


图1-17 用74LS125实现总线传输实验电路

五、实验预习要求

（1）复习TTL集电极开路门和三态门工作原理。

（2）计算实验中各R_L阻值，并从中确定实验所用R_L值（额定值）。

（3）画出用OC与非门实现实验内容2 的逻辑图。

六、实验报告要求

（1）画出实验电路图，并标明有关外接元件值。

（2）整理分析实验结果，总结集电极开路门和三态输出门的优缺点。

（3）分析实验三（1）和（2）中三态门输出电压不同的原因。

（4）思考在使用总线传输时，总线上能不能同时接有OC门与三态输出门？为什么？

实验五 集成逻辑电路的连接和驱动

一、实验目的

(1) 了解TTL门电路的输出特性。

(2) 了解CMOS门电路的输出特性。

(3) 掌握集成逻辑电路相互衔接时应遵守的规则和实际衔接方法。

二、实验原理

1. TTL电路输入输出电路性质

当输入端为高电平时，输入电流是反向二极管的漏电流，电流极小。其方向是从外部流入输入端。当输入端处于低电平时，电流由电源V_{CC}经内部电路流出输入端，电流较大，当与上一级电路衔接时，将决定上级电路应具的负载能力。高电平输出电压在负载不大时为3.5V左右（当Vcc＝5V时）。低电平输出时，允许后级电路灌入电流，随着灌入电流的增加，输出低电平将升高，一般LS系列TTL电路允许灌入8mA电流，即可吸收后级20个LS系列标准门的灌入电流。最大允许低电平输出电压为0.4V。

2. CMOS电路输入输出电路性质

一般CC系列的输入阻抗可高达$10^{10}\Omega$，输入电容在5pF以下，输入高电平通常要求在3.5V以上，输入低电平通常为1.5V以下。因CMOS电路的输出结构具有对称性，故高低电平具有相同的输出能力，负载能力较小，仅可驱动少量的CMOS电路。当输出端负载很轻时，输出高电平将十分接近电源电压；输出低电平时将十分接近地电位。

在高速CMOS电路54/74HC系列中，子系列54/74HCT的输入电平与TTL电路完全相同，因此在相互取代时，不需考虑电平的匹配问题。

3. 集成逻辑电路的衔接

在实际的数字电路系统中总是将一定数量的集成逻辑电路按需要前后连接起来。这时，前级电路的输出将与后级电路的输入相连并驱动后级电路工作。这就存在着电平的配合和负载能力这两个需要妥善解决的问题。可用下列几个表达式来说明连接时所要满足的条件。

V_{OH}（前级）$\geqslant V_{iH}$（后级），V_{OL}（前级）$\leqslant V_{iL}$（后级），

I_{OH}（前级）$\leqslant n\times I_{iH}$（后级），I_{OL}（前级）$\geqslant n\times I_{iL}$（后级），n为后级门的数目。

(1) TTL与TTL的连接

由于电路结构形式相同，对于TTL集成逻辑电路的所有系列，电平配合都比较方便，不需要外接元件就可直接连接，主要的限制是受低电平时负载能力的限制。表1-2列出了74系列TTL电路的扇出系数。

表1-2 74系列芯片扇出系数比较

	74LS00	74ALS00	7400	74L00	74S00
74LS00	20	40	5	40	5
74ALS00	20	40	5	40	5
7400	40	80	10	40	10
74L00	10	20	2	20	1
74S00	50	100	12	100	12

(2) TTL驱动CMOS电路

TTL电路驱动CMOS电路时，由于CMOS电路的输入阻抗高，故驱动电流一般不会受到限制。但在电平配合问题上，低电平是可以的，高电平有困难。因为TTL电路在空载时，输出高电平通常低于CMOS电路对输入高电平的要求，因此，为保证TTL输出高电平，后级的CMOS电路能可靠工作，通常要外接一个上拉电阻R，如图1-18所示，使输出高电平达到3.5V以上，*R*的取值为2kΩ~6.2kΩ较合适，这时TTL后级的CMOS电路的数目可以很多。

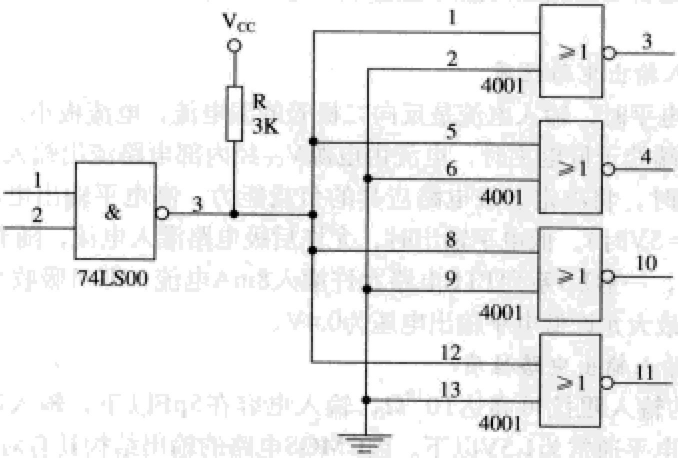


图1-18　TTL电路驱动CMOS电路

(3) CMOS驱动TTL电路

CMOS的输出电平能满足TTL对输入电平的要求，而驱动电流将受限制，主要是低电平时的负载能力。表1-3列出了一般CMOS电路驱动TTL电路的扇出系数，从表中可见，除了74HC系列外的其他CMOS芯片驱动TTL的能力都较低。

表1-3　各种门的驱动能力

	LS-TTL	L-TTL	TTL	ASL-TTL
CC4001B系列	1	2	0	2
MC14001B系列	1	2	0	2
MM74HC及74HCT系列	10	20	2	20

既要使用此系列又要提高其驱动能力时，可采用以下两种方法。

① 采用CMOS专用驱动器，如CC4049和CC4050等芯片。

② 将几个同功能的CMOS电路并联使用，即将其输入端并联，输出端并联（TTL电路是不允许并联的）。

(4) CMOS与CMOS的衔接

CMOS电路之间的连接十分方便，不需另加外接元件。对直流参数来讲，一个CMOS电路可带动的CMOS电路数量是不受限制的。但在实际使用时，应当考虑后级门输入电容对前级门传输速度的影响，电容太大时，传输速度要下降，因此在高速使用时要从负载电容考虑，例如CC4000T系列。CMOS电路在10MHz以上速度运用时应限制在20个门以下。

三、实验设备与器材

(1) 数字逻辑电路实验系统。

（2）数字万用表。

（3）芯片74LS00、74LS04，CC4001、74HC00。

（4）100Ω、470Ω、3kΩ电阻，4.7kΩ可调电位器。

四、实验内容及实验步骤

1. 测试TTL电路74LS00和CMOS电路CC4001的输出特性

（1）测试TTL电路74LS00的输出特性

在数字逻辑电路实验箱IC插座模块处找一个DIP14的插座插上芯片74LS00（CC4001），并在DIP14插座的第7脚接上实验箱的地（GND），第14脚接上电源+5V（V_{CC}）。测试电路如图1-19所示，改变电位器R_W的阻值，从而获得输出特性曲线，R为限流电阻。对图1-19a所示输出高电平进行测试时，应测量到最小允许高电平（2.7V）之间的一系列点；对图1-19b所示输出低电平进行测试时，应测量到最大允许低电平（0.4V）之间的系列点。

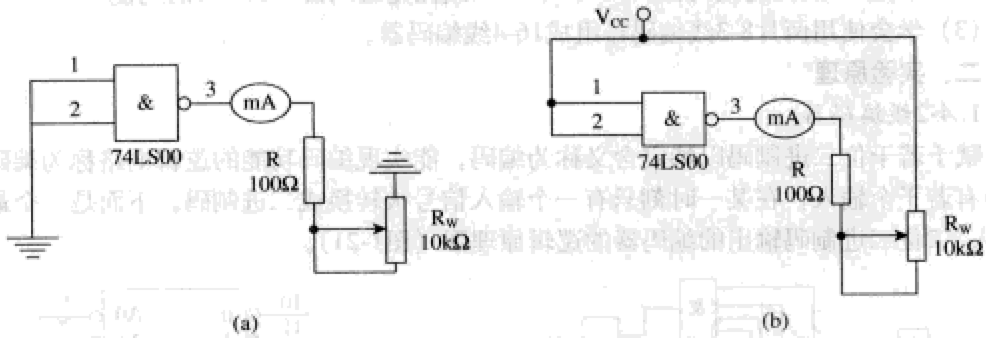


图1-19 与非门输出特性测试电路

（2）测试CMOS电路CC4001的输出特性

同（1）方法，测量出CC4001的输出特性，测试时R取470Ω，R_W取10kΩ不变，V_{CC}为+5V。测试输出高电平时，应测量到最小允许高电平（4.6V）之间的一系列点；测试输出低电平时，应测量到最大允许低电平（0.4V）之间的系列点。

2. TTL电路驱动CMOS电路

用74LS00的一个门来驱动CC4001的四个门，实验电路如图1-18所示。测量连接3kΩ电阻与不连接3kΩ电阻情况下的各芯片逻辑功能及输出的高低电平电压值。

3. CMOS电路驱动TTL电路

电路如图1-20所示，被驱动的电路用74LS04的6个非门输入并联。电路的输入端接逻辑电平输出模块处的输出插口，6个输出分别接逻辑电平显示模块处的输入插口。先用CC4001的一个门来驱动，观测CC4001的输出电平和74LS04的输出逻辑功能。然后将CC4001的其余三个门，逐个并联到第一个门上（输入与输入并联，输出与输出并联），分别观察CMOS的输出电平及74LS04的逻辑功能。最后用1/4 74HC00代替1/4CC4001，测试其输出电平及系统的逻辑功能。

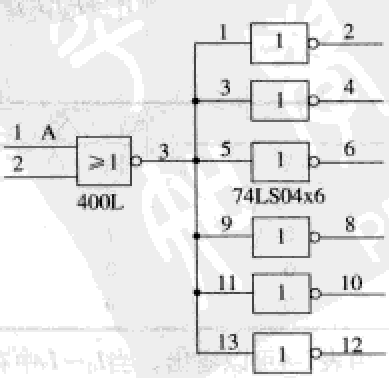


图1-20 CMOS电路驱动TTL电路

五、实验预习要求

（1）自拟各实验记录用的数据表格及逻辑电平记录表格。

（2）熟悉所用集成电路的引脚功能。

六、实验报告要求

（1）整理实验数据，做出输出特性曲线，并加以分析。

（2）通过本次实验，你对不同集成门电路的衔接得出什么结论？

实验六 编码器及其应用

一、实验目的

（1）掌握一种门电路组成编码器的方法。

（2）掌握8-3线优先编码器74LS148，10-4线优先编码器74LS147的功能。

（3）学会使用两片8-3线编码器组成16-4线编码器。

二、实验原理

1. 4-2线编码器

赋予若干位二进制码以特定含义称为编码，能实现编码功能的逻辑电路称为编码器。编码器有若干个输入，在某一时刻只有一个输入信号被转换成二进制码。下面是一个最简单的4输入、2位二进制码输出的编码器的逻辑原理图（图1-21）。

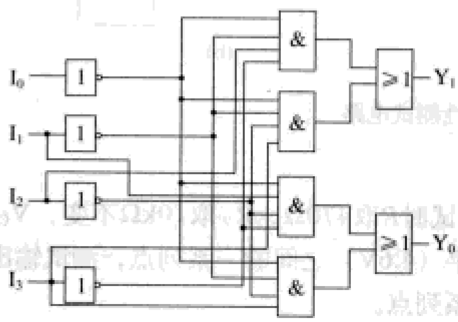


图1-21 4-2线编码器的逻辑原理图

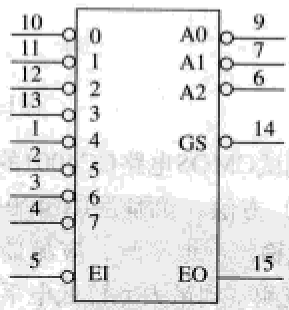


图1-22 74LS148逻辑图

由图可得逻辑表达式为

$$Y_1=\bar{I}_0\bar{I}_1I_2\bar{I}_3+\bar{I}_0\bar{I}_1\bar{I}_2I_3 \qquad Y_0=\bar{I}_0I_1\bar{I}_2\bar{I}_3+\bar{I}_0\bar{I}_1\bar{I}_2I_3$$

表1-4 4-2编码器功能表

输入				输出	
I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

由表1-4可以看出，当I_0～I_3中在某一位输入为1时，输出Y_1 Y_0为相应的代码。例如，当I_1为1时，输出Y_1 Y_0为01。

2. 8-3线优先编码器74LS148

上面的编码电路虽然简单，但有两个缺点。其一是，当I_0为1，$I_1 \sim I_3$都为0和$I_0 \sim I_3$均为0时，输出Y_1 Y_0均为00，这两种情况在实际中必须加以区分；其二是，同时有多个输入被编码时，输出会是混乱的。在实际工作中，同时有多个输入被编码时，必须根据轻重缓急，规定好这些控制对象允许操作的先后顺序，即优先识别。能识别信号的优先级并对其进行编码的逻辑部件称为优先编码器。

编码器74LS148的作用是将输入端0～7这8个状态分别编成二进制码输出，它的功能见下表1-5，它的逻辑图见图1-22。它有8个输入端，3个二进制码输出端，输入使能端EI，输出使能端EO和优先编码工作状态标志GS。优先级分别从I_7至I_0递减。

表1-5　优先编码器74LS148功能表

输入									输出				
$\overline{EI}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$	$\overline{GS}$	EO
H	×	×	×	×	×	×	×	×	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	×	×	×	×	×	×	×	L	L	L	L	L	H
L	×	×	×	×	×	×	L	H	L	L	H	L	H
L	×	×	×	×	×	L	H	H	L	H	L	L	H
L	×	×	×	×	L	H	H	H	L	H	H	L	H
L	×	×	×	L	H	H	H	H	H	L	L	L	H
L	×	×	L	H	H	H	H	H	H	L	H	L	H
L	×	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

注：H表示逻辑高电平，L表示逻辑低电平，×表示逻辑高电平或低电平兼可。

3. 10-4线优先编码器74LS147

74LS147的输出为8421BCD码，它的逻辑图见图1-23，其功能见表1-6。

表1-6　优先编码器74LS147功能表

输入									输出			
$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{8}$	$\overline{9}$	$\overline{D}$	$\overline{C}$	$\overline{B}$	$\overline{A}$
H	H	H	H	H	H	H	H	H	H	H	H	H
×	×	×	×	×	×	×	×	L	L	H	H	L
×	×	×	×	×	×	×	L	H	L	H	H	H
×	×	×	×	×	×	L	H	H	H	L	L	L
×	×	×	×	×	L	H	H	H	H	L	L	H
×	×	×	×	L	H	H	H	H	H	L	H	L
×	×	×	L	H	H	H	H	H	H	L	H	H
×	×	L	H	H	H	H	H	H	H	H	L	L
×	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 数字万用表。

(3) 芯片74LS04、74LS148和74LS20各两片，74LS147、74LS32和74LS08各一片。

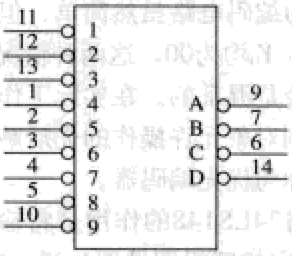


图1-23　74LS147逻辑图

四、实验内容及实验步骤

1. 4-2线编码器

在数字逻辑电路实验箱IC插座模块中找5个DIP14（不够可用DIP16等插座代替）的插座插上芯片74LS04（两片）、74LS20（两片）和74LS32，并将DIP14插座的第7脚接上实验箱的地（GND），第14脚接上电源+5V（V_{CC}）。将输出端Y_0~Y_1分别接2个发光二极管（实验箱主电路板的逻辑电平显示单元），输入端接拨位开关（实验箱主电路板的逻辑电平输出单元）。拨动拨位开关，根据发光二极管显示的变化，逐项验证4-2线编码器的功能。芯片的引脚分配请参考附录1或其他资料。

2. 8-3线优先编码器74LS148

在数字逻辑电路实验箱IC插座模块中找一个DIP16插座插上芯片74LS148，并在DIP16插座的第8脚接上实验箱的地（GND），第16脚接上电源（V_{CC}）。8个输入端0～7接拨位开关（逻辑电平输出），输出端接发光二极管进行显示（逻辑电平显示），其他功能引脚的接法参见附录1或相关资料。

3. 10-4线优先编码器74LS147

测试方法与74LS148类似，只是输入脚与输出脚的个数不同，功能引脚不同。

4. 16-4线编码器

用两片74LS148和一片74LS08组成16位输入、4位二进制码输出的优先编码器（低电平有效），按图1-24逻辑图连线，并验证它的功能。具体的连线方法同样是在IC插座模块上完成，EI_2接低电平，其他输入输出分别接拨位开关（逻辑电平输出）和发光二极管（逻辑电平显示）。

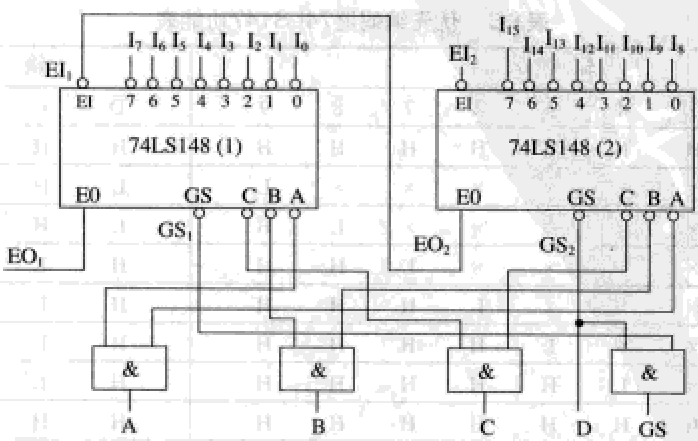


图1-24　16-4线优先编码器原理图

五、实验预习要求

（1）预习编码器的原理。

（2）熟悉所用集成电路的引脚功能。

（3）画好实验所用的表格。

六、实验报告要求

（1）说明74LS148的输入信号EI和输出信号GS、EO的作用。

（2）分析16-4线优先编码器的工作原理，并自制表格，根据实验结果完成16-4线优先编码器的功能表。

实验七 译码器及其应用

一、实验目的

（1）掌握3-8线译码器、4-10线译码器的逻辑功能和使用方法。

（2）掌握用两片3-8线译码器连成4-16线译码器的方法。

（3）掌握使用74LS138实现逻辑函数和做数据分配器的方法。

二、实验原理

译码是编码的逆过程，它的功能是辨别具有特定含义的二进制码，并将其转换成控制信号。具有译码功能的逻辑电路称为译码器。译码器在数字系统中有广泛的应用，不仅用于代码的转换和终端的数字显示，还用于数据分配、存贮器寻址和组合控制信号等。不同的功能可选用不同种类的译码器。

图1-25表示二进制译码器的一般原理图。它具有n个输入端，2^n个输出端和一个使能输入端。在使能输入端为有效电平时，对应每一组输入代码，只有其中一个输出端为有效电平，其余输出端则为无效电平。每一个输出所代表的函数对应于n个输入变量的最小项。二进制译码器实际上也是负脉冲输出的脉冲分配器，若利用使能端中的一个输入端输入数据信息，器件就成为一个数据分配器（又称为多路数据分配器）。

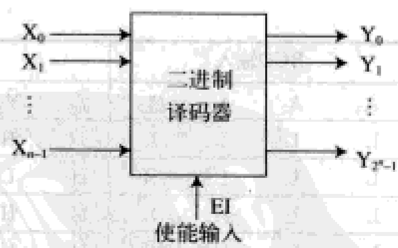


图1-25 二进制译码器的一般原理图

1. 3-8线译码器74LS138

它有三个地址输入端A、B、C，它们共有8种状态的组合，即可译出8个输出信号Y_0~Y_7。另外它还有三个使能输入端E_1、E_2、E_3。它的功能如表1-7所示，引脚排列见图1-26。

表1-7 74LS138的功能表

输入						输出							
E_3	$\bar{E}_1$	$\bar{E}_2$	C	B	A	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$
×	H	×	×	×	×	H	H	H	H	H	H	H	H
×	×	H	×	×	×	H	H	H	H	H	H	H	H
L	×	×	×	×	×	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H

（续）

输入						输出							
E_3	$\bar{E}_1$	$\bar{E}_2$	C	B	A	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

2. 4-10线译码器74LS42

4-10线译码器74LS42的引脚排列见图1-27，功能如表1-8所示。

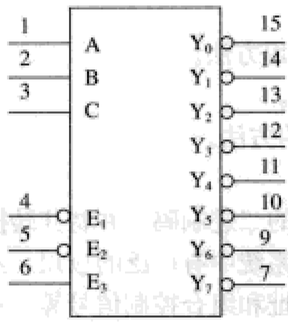


图1-26 74LS138引脚排列图

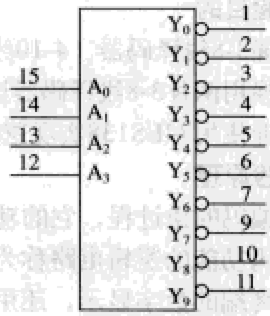


图1-27 74LS42的引脚排列图

表1-8 74LS42功能表

BCD输入				输出									
A_3	A_2	A_1	A_0	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$	$\bar{Y}_8$	$\bar{Y}_9$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）数字万用表。

（3）双踪示波器。

（4）芯片74LS138两片，74LS42和74LS20各一片。

四、实验内容及实验步骤

1. 74LS138译码器逻辑功能测试

在数字逻辑电路实验箱IC插座模块中找一个DIP16的插座插上芯片74LS138，并将DIP16插座的第8脚接上实验箱的地（GND），第16脚接上电源+5V（VCC）。将74LS138的输出端Y0~Y7分别接到8个发光二极管上（逻辑电平显示单元），输入端接拨位开关输出（逻辑电平输出单元），逐次拨动开关，根据发光二极管显示的变化，测试74LS138的逻辑功能。

2. 74LS42译码器逻辑功能测试

测试方法与74LS138类似，只是输入脚与输出脚的个数不同，功能引脚不同。

3. 两片74LS138组合成4-16线译码器

按图1-28连线。

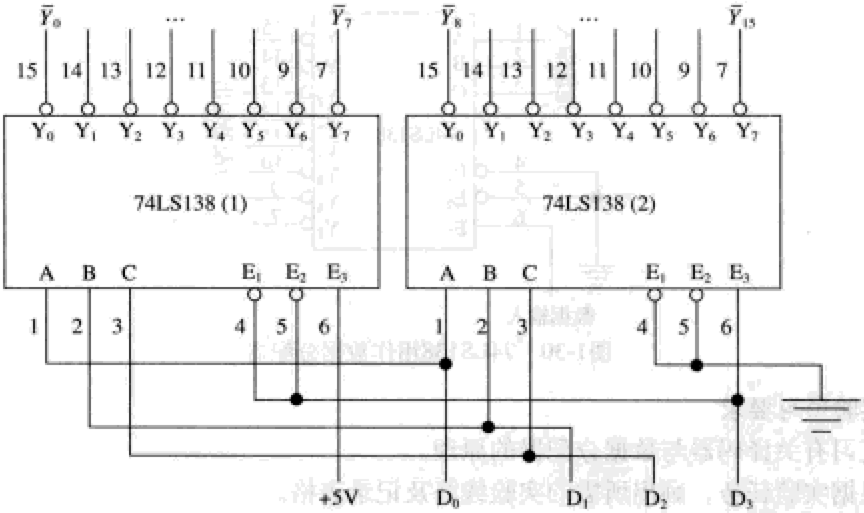


图1-28 两片74LS138组合成4-16线译码器

将16个输出端接逻辑电平显示（发光二极管），4个输入端接逻辑电平输出（拨位开关），逐项测试电路的逻辑功能。

4. 用74LS138实现逻辑函数和做数据分配器

（1）实现逻辑函数

一个3-8线译码器能产生3变量函数的全部最小项，利用这一点能够很方便地实现3变量逻辑函数。图1-29实现了 $F=\overline{XYZ}+\overline{X}Y\overline{Z}+X\overline{Y}Z+XYZ$ 功能输出。

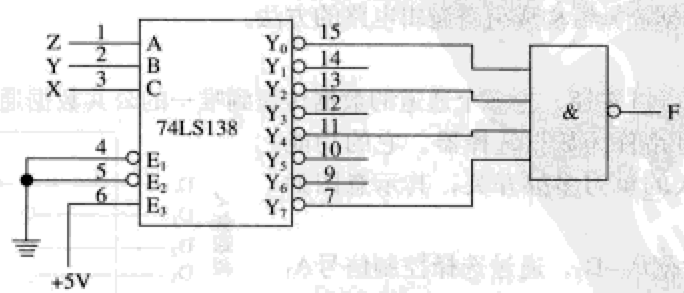


图1-29 用74LS138实现逻辑函数

验证电路的功能是否与逻辑函数相一致。具体的接线方法是在IC插槽部分找相应插槽插上芯片74LS138和74LS20，X、Y、Z三个输入端接拨位开关（逻辑电平输出），F接发光二极

管（逻辑电平显示），拨动拨位开关，观察发光二极管的发光情况。

（2）用作数据分配器

若在E3端输入数据信息，$\overline{E1}=\overline{E2}=0$，地址码所对应的输出是E3数据的反码，若从$\overline{E2}$端输入数据信息，令$\overline{E3}=1$，$\overline{E1}=0$，地址码所对应的输出是$\overline{E2}$端数据信息的原码。若输入信息是时钟脉冲，则数据分配器便成为时钟脉冲分配器。

取时钟脉冲CP的频率约为10kHz，要求分配器输出端$\overline{Y}_0\sim\overline{Y}_7$的信号与CP输入信号同相。参照图1-30，画出分配器的实验电路，用示波器观察和记录在地址端CBA分别取000~111这8种不同状态时$\overline{Y}_0\sim\overline{Y}_7$端的输出波形，注意输出波形与CP输入波形之间的相位关系。

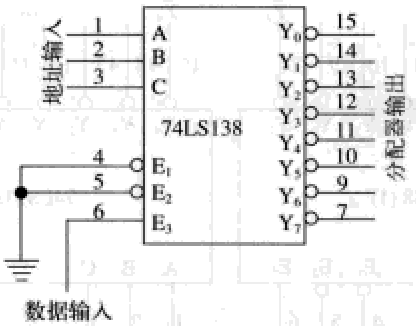


图1-30 74LS138用作数据分配器

五、实验预习要求

（1）复习有关译码器与数据分配器的原理。

（2）根据实验任务，画出所需的实验线路及记录表格。

六、实验报告要求

（1）画出实验线路，把观察到的波形画在坐标上，并标上相应的地址码。

（2）对实验结果进行分析和讨论。

实验八 数据选择器及其应用

一、实验目的

（1）掌握数据选择器的逻辑功能和使用方法。

（2）学习用数据选择器构成组合逻辑电路的方法。

二、实验原理

数据选择是指经过选择，把多个通道的数据传送到唯一的公共数据通道上去。实现数据选择功能的逻辑电路称为数据选择器。它的功能相当于一个多输入的单刀多掷开关，其示意图如图1-31所示。

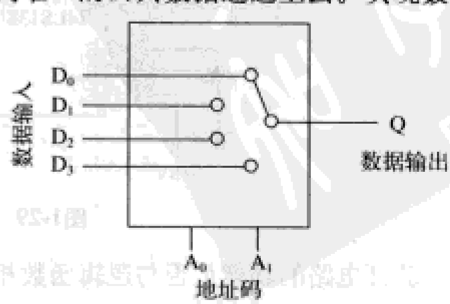


图1-31 4选1数据选择器示意图

图中有四路数据$D_0\sim D_3$，通过选择控制信号A_1和A_0（地址码）从四路数据中选中一路数据送至输出端Q。

1. 八选一数据选择器74LS151

74LS151是一种典型的集成电路数据选择器，

它有3个地址输入端CBA，可选择D0~D7这8个数据源，具有两个互补输出端、同相输出端Y和反相输出端WN。其引脚图如图1-32所示，功能表如表1-9所示，功能表中H表示逻辑高电平，L表示逻辑低电平，×表示逻辑高电平或低电平。

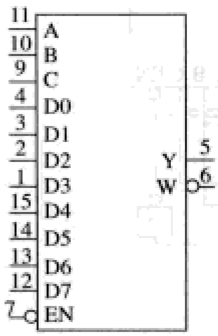


图1-32 74LS151的引脚图

表1-9 74LS151的功能表

C	B	A	$\overline{EN}$	Y	$\overline{W}$
×	×	×	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

2. 双四选一数据选择器74LS153

74LS153数据选择器有两个完全独立的4选1数据选择器，每个数据选择器有4个数据输入端I0~I3，2个地址输入端S0和S1，1个使能控制端$\bar{E}$和一个输出端Z，它们的功能见表1-10，引脚逻辑图如图1-33所示。其中，$\overline{EA}$、$\overline{EB}$（1脚、15脚）分别为A路和B路的选通信号，I0、I1、I2、I3为4个数据输入端，ZA（7脚）、ZB（9脚）分别为两路的输出端，S0（14脚）、S1（2脚）为地址信号，8脚为GND，16脚为V_{CC}。

表1-10 74LS153真值表

S_0	S_1	$\bar{E}$	I_0	I_1	I_2	I_3	Z
×	×	H	×	×	×	×	L
L	L	L	L	×	×	×	L
L	L	L	H	×	×	×	H
H	L	L	×	L	×	×	L
H	L	L	×	H	×	×	H
L	H	L	×	×	L	×	L
L	H	L	×	×	H	×	H
H	H	L	×	×	×	L	L
H	H	L	×	×	×	H	H

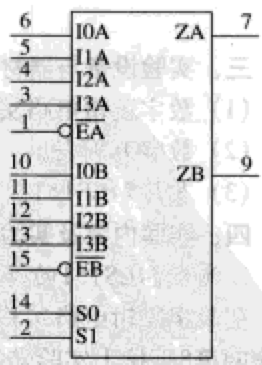


图1-33 74LS153引脚逻辑图

3. 用74LS151组成16选1数据选择器

用低三位$A_2A_1A_0$作为每片74LS151的片内地址码，用高位A_3作为两片74LS151的片选信号。当A_3=0时，选中74LS151（1）工作，74LS151（2）禁止；当A_3=1时，选中74LS151（2）工作，74LS151（1）禁止，如图1-34所示。

4. 数据选择器的应用

用74LS153实现逻辑函数 $Y=\bar{A}BC+A\bar{B}C+AB\bar{C}+ABC$ 。

函数Y有三个输入变量A、B、C，而数据选择器有两个地址输入端$\overline{S_1}$、$\overline{S_0}$，少于3个。可考虑把数据输入端作为变量之一，图1-35实现了函数Y的功能。即将A、B分别接选择器的地址端$\overline{S_1}$和$\overline{S_0}$，并令I_0=0，I_1=I_2=C。

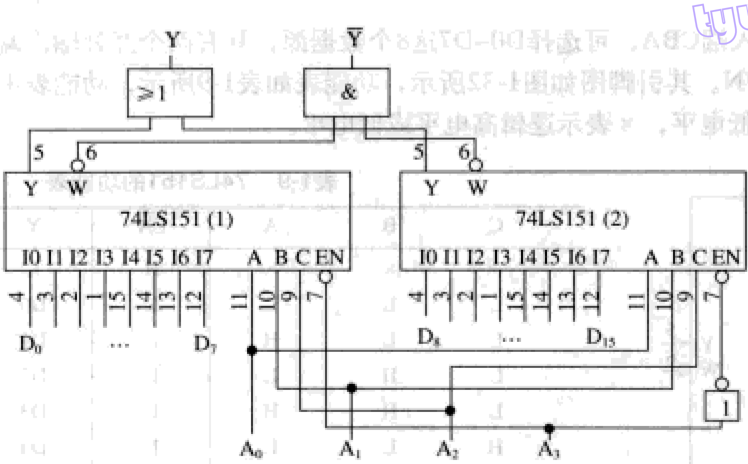


图1-34 用74LS151组成16选1数据选择器

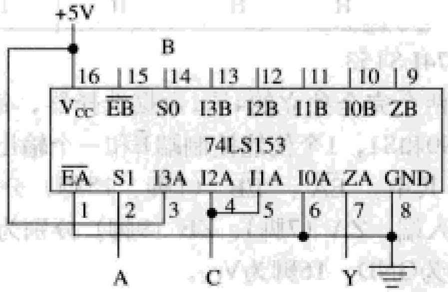


图1-35 用74LS153实现逻辑函数

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 数字万用表。

(3) 芯片74LS151、74LS153、74LS04、74LS08和74LS32。

四、实验内容及实验步骤

1. 测试74LS151的逻辑功能

在数字逻辑电路实验箱IC插座模块中找一个DIP16的插座插上芯片74LS151，并将DIP16插座的第8脚接上实验箱的地（GND），第16脚接上电源（V_{CC}）。将74LS151的输出端Z接到发光二极管上（逻辑电平显示单元），自己接线，按74LS151的真值表逐项进行测试，记录测试结果。

2. 测试74LS153的逻辑功能

测试方法与步骤同上，记录测试结果。

3. 用两片74LS151组成16选1数据选择器

按图1-34接线，自己设计，灵活利用逻辑电平输出拨位开关。记录结果并分析。

4. 用74LS153实现逻辑函数

参考图1-35，分析该图的实现方法，并验证其逻辑功能。现要求实现 $F=\overline{A}B+A\overline{B}$ ，自己写出设计过程，画出接线图，并验证其逻辑功能。

五、实验预习要求

(1) 复习数据选择器的工作原理。

(2) 用数据选择器对实验内容中的各函数式进行预设计。

(3) 思考能否用数据选择器实现全加器功能。

六、实验报告要求

(1) 用数据选择器对实验内容进行设计，写出设计全过程，画出接线图，并进行逻辑功能测试。

(2) 写一篇有关编码器、译码器和数据选择器实验的收获与体会的文章。

实验九 加法器与数值比较器

一、实验目的

(1) 掌握半加器和全加器的工作原理。

(2) 掌握数值比较器的工作原理。

(3) 掌握4位数值比较器74LS85的逻辑功能。

二、实验原理

1. 半加器

半加器的真值表如表1-11所示。由表可以看出这种加法运算只考虑了两个加数本身，而没有考虑由低位来的进位，所以称为半加。

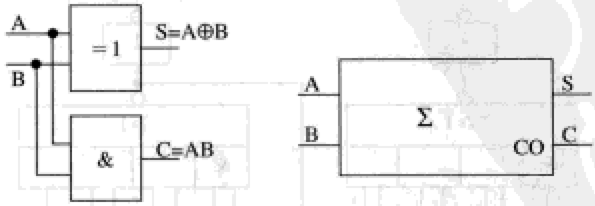
表1-11 半加器的真值表

被加数A	加数B	和数S	进位数C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

由真值表可得

$$S=\overline{A}B+A\overline{B}$$
$$C=AB$$

用异或门和与门组成的半加器的原理图如图1-36所示。



(a) 由异或门和与门组成　(b) 半加器的符号

图1-36 半加器

2. 全加器

全加器能将加数、被加数和低位来的进位信号相加，并根据求和的结果给出该位的进位

信号。根据全加器的功能，可列出它的真值表，如表1-12所示。其中，C_{i-1}为相邻低位来的进位数，S_i为本位和数（称为全加和），C_i为向相邻高位的进位数。

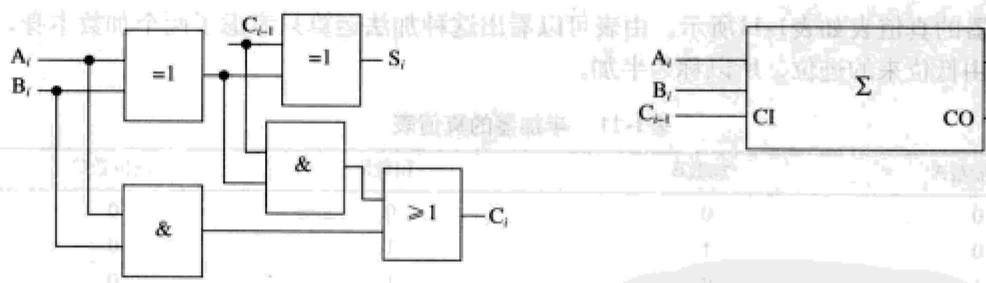
表1-12 全加器的真值表

A_i	B_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

由全加器的真值表可以写出S_i和C_i的逻辑表达式

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$
$$C_i = A_iB_i + (A_i \oplus B_i)C_{i-1}$$

它的原理图如图1-37所示。



(a) 由异或门、或门和与门组成　　(b) 全加器的符号

图1-37 全加器

用非门和与或非门组成的全加器逻辑电路如图1-38所示，读者可自行分析输入输出的关系，并通过实验加以验证。

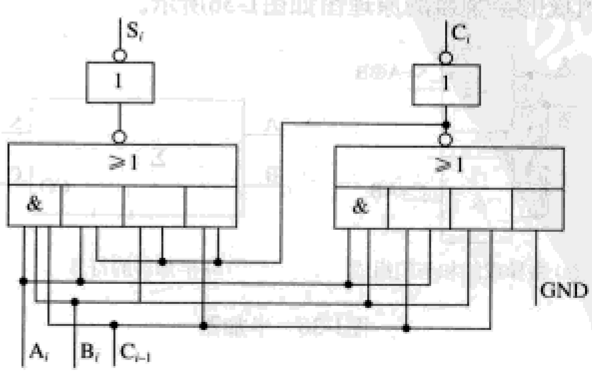


图1-38 用非门和与或非门组成的全加器

3. 数值比较器的原理

在数字系统中，常常要比较两个数的大小。数值比较器就是对两数A和B进行比较，以判断其大小的逻辑电路。比较结果有A>B、A<B、A=B三种情况。表1-13和图1-39分别是最简单的一位数值比较器的真值表和逻辑电路图。

表1-13 一位数值比较器的真值表

输入		输出		
A	B	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

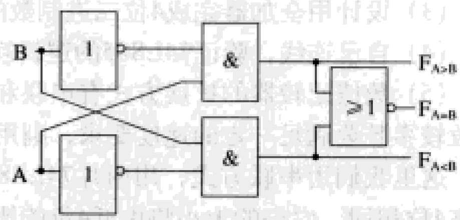


图1-39 一位数值比较器的逻辑电路图

对于多位的情况，一般而言，先比较高位。当高位不等时，两个数的比较结果就是高位的比较结果，当高位相等时，两数的比较结果由低位决定。

集成数值比较器74LS85

集成数值比较器74LS85是四位数值比较器，它的引脚图和真值表分别如图1-40和表1-14所示。



图1-40 74LS85的引脚图

其中，10脚、12脚、13脚、15脚和1脚、9脚、11脚、14脚是输入端，5脚、6脚、7脚为输出端，2脚、3脚、4脚为级联输入端，8脚为地，16脚为电源。

表1-14 74LS85的真值表

输入							输出		
A_3，B_3	A_2，B_2	A_1，B_1	A_0，B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
$A_3>B_3$	×	×	×	×	×	×	H	L	L
$A_3<B_3$	×	×	×	×	×	×	L	H	L
$A_3=B_3$	$A_2>B_2$	×	×	×	×	×	H	L	L
$A_3=B_3$	$A_2<B_2$	×	×	×	×	×	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	×	×	×	×	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	×	×	×	×	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	×	×	×	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	×	×	×	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	L	L	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	H	L	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	×	×	H	L	L	H
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	H	L	L	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	L	L	H	H	L

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）数字万用表。

（3）芯片74LS85×2、74LS86、74LS04、74LS08、74LS32和74LS64×2。

四、实验内容及实验步骤

(1) 在数字逻辑电路实验箱IC插座模块中，插上实验需要的芯片，用门电路组成一个半加器，连线并验证其逻辑功能，自拟真值表，并将实验结果填入表中。

(2) 用门电路组成一个全加器，连线并验证其逻辑功能，自拟真值表，并将实验结果填入表中，然后与逻辑表达式比较。

(3) 设计用全加器完成4位二进制数的相加，验证其逻辑功能。

(4) 自己连线，验证74LS85的逻辑功能。

(5) 数值比较器的扩展方式有串联和并联两种。一般位数较少的话，用串联方式；如果位数较多且要满足一定的速度要求，则用并联方式。

这里我们用串联方式，用两片74LS85组成8位数值比较器。我们知道，对于两个8位数，若高4位相同，它们的大小将由低4位的比较结果确定。因此，低4位的比较结果作为高4位的条件，即低4位比较器的输出端应分别与高4位比较器的$I_{A>B}$、$I_{A<B}$和$I_{A=B}$端连接，见图1-41。

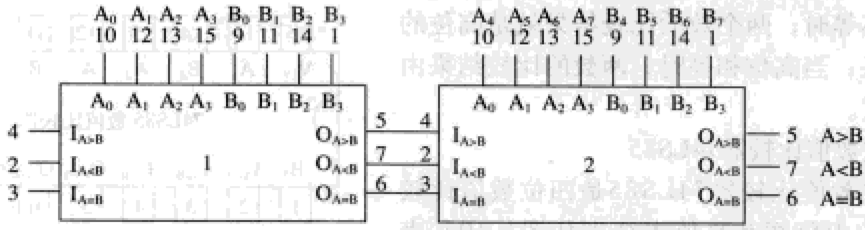


图1-41 用两片74LS85组成8位数值比较器

具体的实验方法为，在IC插座模块上插上两片74LS85（注意芯片插在DIP16的插座上），按照图1-41连线，实现8位数值比较器功能。

五、实验预习要求

(1) 认真复习半加器、全加器和数值比较器的工作原理。

(2) 自己查找资料学习如何使用74LS85。

(3) 实验前，画好实验用的电路图和表格。

六、实验报告要求

(1) 参考教材及有关资料，设计简单的半减器和全减器，画出电路图和真值表，并验证其逻辑功能。

(2) 用简单的逻辑门设计一个2位二进制数值比较器，画出逻辑电路图，将实验结果填入自制的表中。

(3) 思考如何用全加器实现多位数的相加。

实验十 组合逻辑电路的设计与冒险的测试

一、实验目的

(1) 掌握组合逻辑电路的分析与设计方法。

(2) 加深对基本门电路的理解。

二、实验原理

1. 组合电路概述

组合电路是最常用的逻辑电路，可以用一些常用的门电路来组合完成具有其他功能的门电路。例如，根据与门的逻辑表达式 $Z=AB=\overline{\overline{A}+\overline{B}}$ 可知，用两个非门和一个或非门可以组合成一个与门。当然，还可以组合成更复杂的逻辑关系。

2. 分析组合逻辑电路的一般步骤

分析组合逻辑电路的一般步骤如下。

（1）由逻辑图写出各输出端的逻辑表达式。

（2）化简和变换各逻辑表达式。

（3）列出真值表。

（4）根据真值表和逻辑表达式对逻辑电路进行分析，最后确定其功能。

3. 设计组合逻辑电路的一般步骤

与上面相反，设计组合逻辑电路的一般步骤为下面几步。

（1）根据任务的要求，列出真值表。

（2）用卡诺图或代数化简法求出最简的逻辑表达式。

（3）根据表达式，画出逻辑电路图，用标准器件构成电路。

（4）最后，用实验来验证设计的正确性。

4. 组合逻辑电路的设计举例

例：用“与非门”设计一个表决电路。当4个输入端中有3个或4个“1”时，输出端才为“1”。

设计步骤：

根据题意，列出真值表如表1-15所示，再填入卡诺图表1-16中。

表1-15　表决电路的真值表

D	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
A	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Z	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1

表1-16　表决电路的卡诺图

BC \ DA	00	01	11	10
00				
01			1	
11		1	1	1
10			1	

然后，由卡诺图得出逻辑表达式，并演化成“与非”的形式

$$Z=ABC+BCD+CDA+ABD$$

$$=\overline{\overline{ABD}\cdot\overline{BCD}\cdot\overline{ACD}\cdot\overline{ABC}}$$

最后，画出用“与非门”构成的逻辑电路如图1-42所示。

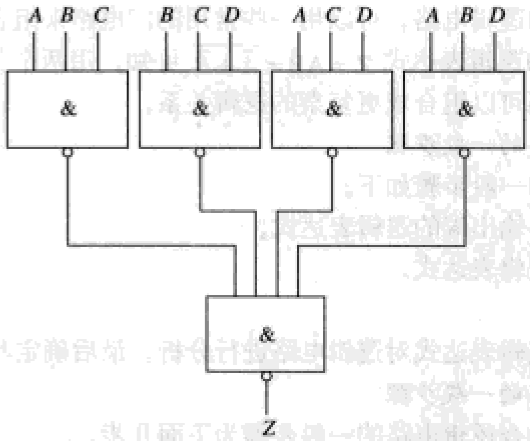


图1-42　表决电路原理图

输入端接至逻辑开关（拨位开关）输出插口，输出端接逻辑电平显示端口，自拟真值表，逐次改变输入变量，验证逻辑功能。

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）数字万用表。

（3）芯片74LS00（×2）、74LS04、74LS10、74LS20、74LS85和74LS86。

四、实验内容及实验步骤

（1）完成组合逻辑电路设计中的例子。

（2）设计一个保险箱用的4位数字代码锁，该锁有规定的地址代码A、B、C、D 4个输入端和一个开箱钥匙孔信号E的输入端，锁的代码由实验者自编。当用钥匙开箱时，如果输入的4个地址代码正确，保险箱被打开；否则，电路将发出警报（可用发光二极管亮表示）。

（3）用与非门74LS00和异或门74LS86设计一可逆的4位码变换器。要求如下。

① 当控制信号C=1时，它将8421码转换成为格雷码；当控制信号C=0时，它将格雷码转换成为8421码。

② 写出设计步骤，列出码变换关系真值表并画出逻辑电路图。

③ 连接电路并测试逻辑电路的功能。

（4）将74LS04中的三个反相器串接在一起（前级的输出作为下一级的输入）。第1级反相器的输入接1MHz脉冲源。将第1级反相器的输入和第3级反相器的输出分别作为74LS00中一个与非门的输入，用示波器观测与非门的输出。

（5）将74LS04中的5个反相器串接在一起。第1级反相器的输入接1MHz脉冲源。将第1级反相器的输入和第5级反相器的输出分别作为74LS00中一个与非门的输入，用示波器观测与非门的输出。

五、实验预习要求

（1）复习各种基本门电路的使用方法。

（2）实验前，画好实验用的电路图和表格。

（3）自己参考有关资料画出实验内容（2）和（3）中的原理图，找出实验要使用的芯片，以备实验时用。

六、实验报告要求

(1) 将实验结果填入自制的表格中，验证设计是否正确。

(2) 总结组合逻辑电路的分析与设计方法。

(3) 画出实验内容 (4) 和 (5) 中在与非门输出端观测到的波形图，分析波形图上冒险现象产生的原因。

实验十一　触发器及其应用

一、实验目的

(1) 掌握基本RS触发器、JK触发器、T触发器和D触发器的逻辑功能。

(2) 掌握集成触发器的功能和使用方法。

(3) 熟悉触发器之间相互转换的方法。

二、实验原理

触发器是能够存储1位二进制码的逻辑电路，它有两个互补输出端，其输出状态不仅与输入有关，而且还与原先的输出状态有关。触发器有两个稳定状态，用以表示逻辑状态"1"和"0"，在一定的外界信号作用下，可以从一个稳定状态翻转到另一个稳定状态，它是一个具有记忆功能的二进制信息存储器件，是构成各种时序电路的最基本逻辑单元。

1. 基本RS触发器

图1-43为由两个与非门交叉耦合构成的基本RS触发器，它是无时钟控制低电平直接触发的触发器。基本RS触发器具有置"0"、置"1"和保持三种功能。通常称 $\overline{S}$ 为置"1"端，因为 $\overline{S}=0$ 时触发器被置"1"；$\overline{R}$ 为置"0"端，因为 $\overline{R}=0$ 时触发器被置"0"。当 $\overline{S}=\overline{R}=1$ 时状态保持，当 $\overline{S}=\overline{R}=0$ 时为不定状态，应当避免这种状态。

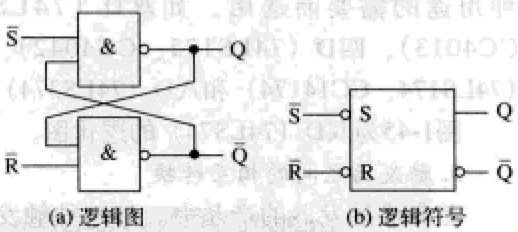


图1-43　二与非门组成的基本RS触发器

基本RS触发器也可以用两个"或非门"组成，此时为高电平有效。

基本RS触发器的逻辑符号见图1-43b，二输入端的边框外侧都画有小圆圈，这是因为置1与置0都是低电平有效。

2. JK触发器

在输入信号为双端的情况下，JK触发器是功能完善、使用灵活和通用性较强的一种触发器。本实验采用74LS112双JK触发器，是下降边沿触发的边沿触发器。逻辑图如图1-44所示。JK触发器的状态方程为

$$Q^{n+1}=J\overline{Q}^n+\overline{K}Q^n$$

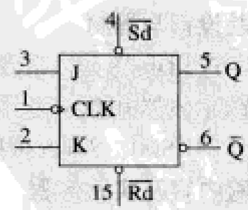


图1-44　JK触发器的逻辑图

其中，J和K是数据输入端，是触发器状态更新的依据，若J、K有两个或两个以上输入端时，组成"与"的关系。Q和 $\overline{Q}$ 为两个互补输入端。通常把Q=0、$\overline{Q}=1$ 的状态定为触发器"0"状态；而把Q=1，$\overline{Q}=0$ 定为"1"状态。

JK触发器常用作缓冲存储器、移位寄存器和计数器。

74LS73A也是双JK触发器，其功能与74LS112相同，只是不带置位端 $\overline{Sd}$ 。

3. T触发器

在JK触发器的状态方程中，令J=K=T则变换为

$$Q^{n+1}=T\overline{Q^n}+\overline{T}Q^n$$

这就是T触发器的特性方程。由上式有

$$当T=1时，\ Q^{n+1}=\overline{Q^n}$$

$$当T=0时，\ Q^{n+1}=Q^n$$

即当T=1时，为翻转状态；当T=0时，为保持状态。

4. D触发器

在输入信号为单端的情况下，D触发器用起来更为方便，其状态方程为

$$Q^{n+1}=D$$

其输出状态的更新发生在CP脉冲的上升沿，故又称为上升沿触发的边沿触发器，触发器的状态只取决于时钟到来前D端的状态，D触发器的应用很广，可用于数字信号的寄存、移位寄存、分频和波形发生等。它有很多型号可供各种用途的需要而选用。如双D（74LS74、CC4013）、四D（74LS175、CC4042）、六D（74LS174、CC14174）和八D（74LS374）等。

图1-45为双D（74LS74）的逻辑图。

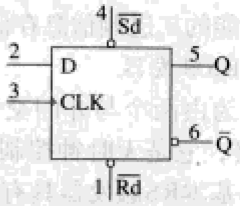


图1-45 D触发器的逻辑图

5. 触发器之间的相互转换

在集成触发器的产品中，每一种触发器都有自己固定的逻辑功能。但是可以利用转换的方法获得具有其他功能的触发器。例如将JK触发器的J、K两端接在一起，并认它为T端，就得到所需的T触发器。

JK触发器也可以转换成为D触发器，如图1-46所示。

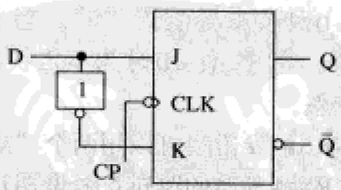


图1-46 JK触发器转换成为D触发器

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）双踪示波器和数字万用表。

（3）芯片74LS00、74LS74和74LS112（或74LS73A）。

四、实验内容及实验步骤

1. 测试基本RS触发器的逻辑功能

按图1-43，用两个与非门组成基本RS触发器，输入端 $\overline{S}$ 、$\overline{R}$ 接逻辑电平输出插孔（拨位开关输出端），输出端Q和 $\overline{Q}$ 接逻辑电平显示输入插孔（发光二极管输入端），测试它的逻辑功能并画出真值表将实验结果填入表内。（测量不定状态时，建议将 $\overline{S}$ 和 $\overline{R}$ 端并联，连接到单次脉冲上进行控制。）

2. 测试JK触发器74LS112（或74LS73A）的逻辑功能

（提示：74LS73的引脚11是GND，引脚4是Vcc）

（1）测试JK触发器的复位和置位功能

任取一个JK触发器，$\overline{Sd}$、$\overline{Rd}$、J、K端接逻辑电平输出插孔，CP接单次脉冲源，输出端Q和$\overline{Q}$接逻辑电平显示输入插孔。要求改变$\overline{Sd}$、$\overline{Rd}$（J、K和CP处于任意状态），并在$\overline{Rd}=0$（$\overline{Sd}=1$）或$\overline{Rd}=1$（$\overline{Sd}=0$）作用期间任意改变J、K和CP的状态，观察Q和$\overline{Q}$的状态，自拟表格并记录之。

（2）测试JK触发器的逻辑功能

令$\overline{Sd}=\overline{Rd}=1$，不断改变$J$、$K$和$CP$的状态，观察$Q$和$\overline{Q}$的状态变化，观察触发器状态更新是否发生在$CP$的下降沿，记录之。

（3）将JK触发器的J、K端连在一起，构成T触发器

在CP端输入1kHz连续脉冲，观察Q端的变化，用双踪示波器观察CP、Q和$\overline{Q}$的波形，注意相位关系，描绘之。

（4）JK触发器转换成D触发器

按图1-46连线，方法与步骤同上，测试D触发器的逻辑功能，并画出真值表，将实验结果填入表内。

3. 基本RS触发器的应用

图1-47是由基本RS触发器构成的去抖动电路开关，它是利用基本RS触发器的记忆功能来消除开关振动带来的影响。参考有关资料分析其工作原理，自己在实验电路板上搭建电路来验证该去抖动电路的功能。

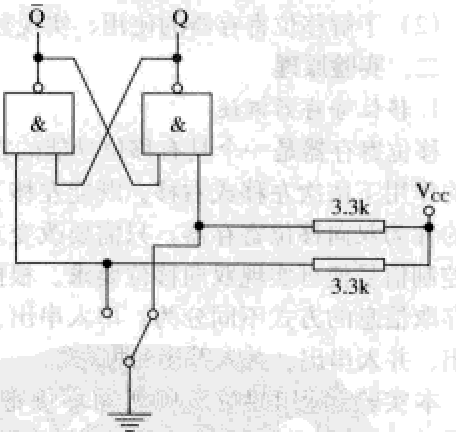


图1-47 去抖动电路图

4. 测试双D触发器74LS74的逻辑功能

（1）测试D触发器的复位和置位功能

测试方法与步骤同实验内容2（1），只是它们的功能引脚不同，相关的引脚分布参见附录1，自拟表格记录。

（2）测试D触发器的逻辑功能

按表1-17要求进行测试，并观察触发器状态是否发生在CP脉冲的上升沿（即由0变1），记录之。

表 1-17

D	CP	Q^{n+1}	
		$Q^n=0$	$Q^n=1$
0	0变1		
	1变0		
1	0变1		
	1变0		

五、实验预习要求

(1) 复习有关触发器内容，熟悉有关器件的引脚分配。

(2) 列出各触发器功能测试表格。

(3) 参考有关资料，查看74LS112（或74LS73A）和74LS74的逻辑功能。

六、实验报告要求

(1) 列表整理各类触发器的逻辑功能。

(2) 总结观察到的波形，说明触发器的触发方式。

(3) 利用普通的机械开关组成的数据开关所产生的信号，是否可以作为触发器的时钟脉冲信号，为什么？是否可以作为触发器的其他输入端的信号，又是为什么？

(4) 思考为什么图1-47所示的去抖动电路能去抖动。

实验十二 移位寄存器及其应用

一、实验目的

(1) 掌握4位双向移位寄存器的逻辑功能与使用方法。

(2) 了解移位寄存器的使用，实现数据的串行、并行转换和构成环形计数器。

二、实验原理

1. 移位寄存器概述

移位寄存器是一个具有移位功能的寄存器。移位是指寄存器中所存的代码能够在移位脉冲的作用下依次左移或右移。既能左移又能右移的称为双向移位寄存器，只需要改变左右移的控制信号便可实现双向移位要求。根据寄存器存取信息的方式不同分为：串入串出、串入并出、并入串出、并入并出4种形式。

本实验选用的4位双向通用移位寄存器，型号为74LS194或CC40194，两者功能相同，可互换使用，其引脚排列如图1-48所示。

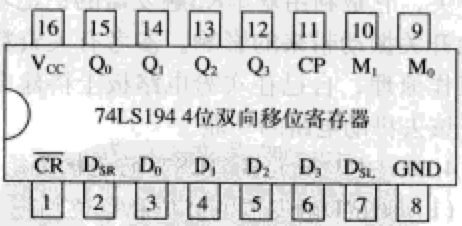


图1-48 74LS194（或CC40194）的引脚排列图

其中，D_{SR}为右移串行输入端，D_{SL}为左移串行输入端。其功能如表1-18所示。

表1-18 74LS194功能表

输入										输出				功能
清零	控制信号		串行输入		时钟	并行输入								
$\overline{CR}$	M_1	M_0	左移D_{SL}	右移D_{SR}	CP	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3	
0	×	×	×	×	×	×	×	×	×	0	0	0	0	异步清零
1	×	×	×	×	0/1	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n	保持
1	0	0	×	×	×	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n	
1	0	1	×	1	↑	×	×	×	×	1	Q_0^n	Q_1^n	Q_2^n	右移
1	0	1	×	0	↑	×	×	×	×	0	Q_0^n	Q_1^n	Q_2^n	
1	1	0	1	×	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	1	左移
1	1	0	0	×	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	0	
1	1	1	×	×	↑	d_0	d_1	d_2	d_3	d_0	d_1	d_2	d_3	并入

2. 移位寄存器的应用

移位寄存器的应用很广，可构成移位寄存器型计数器、顺序脉冲发生器和串行累加器；可用于数据转换，即把串行数据转换为并行数据，或把并行数据转换为串行数据等。

（1）环形计数器

把移位寄存器的输出反馈到它的串行输入端，就可以进行循环移位，如图1-49所示。

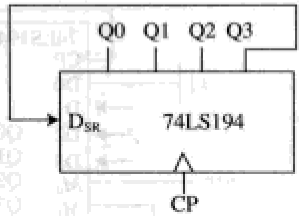


图1-49 环形计数器

将输出端Q3与输入端D_{SR}相连后，在时钟脉冲的作用下Q0Q1Q2Q3将依次右移。同理，将输出端Q0与输入端D_{SL}相连后，在时钟脉冲的作用下Q0Q1Q2Q3将依次左移。

（2）实现数据串行/并行转换

① 串行/并行转换器

串行/并行转换是指，串行输入的数据经过转换电路之后变成并行输出。图1-50是用两片74LS194构成的7位串行/并行转换电路。

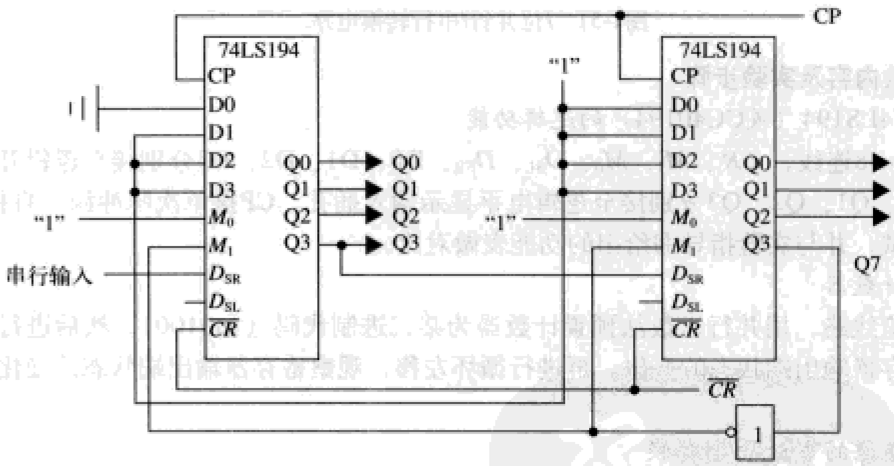


图1-50 7位串行/并行转换电路

电路中M_0端接高电平1，M_1受$Q7$控制，两片寄存器连接成串行输入右移工作模式。$Q7$是转换结束标志。当$Q7=1$时，M_1为0，使之成为$M_1M_0=01$的串入右移工作方式。当$Q7=0$时，M_1为1，有$M_1M_0=11$，则串行送数结束，标志着串行输入的数据已转换成并行输出。

② 并行/串行转换器

并行/串行转换是指，并行输入的数据经过转换电路之后变成串行输出。图1-51是用两片74LS194构成的7位并行/串行转换电路。与图1-50相比，它多了两个与非门，而且还多了一个转换启动信号（负脉冲或低电平），工作方式同样为右移。

对于中规模的集成移位寄存器，其位数往往以4位居多，当所需要的位数多于4位时，可以把几片集成移位寄存器用级连的方法来扩展位数。

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）双踪示波器和数字万用表。

(3) 芯片74LS00、74LS04、74LS30（8输入与非门）和74LS194（或CC40194）。

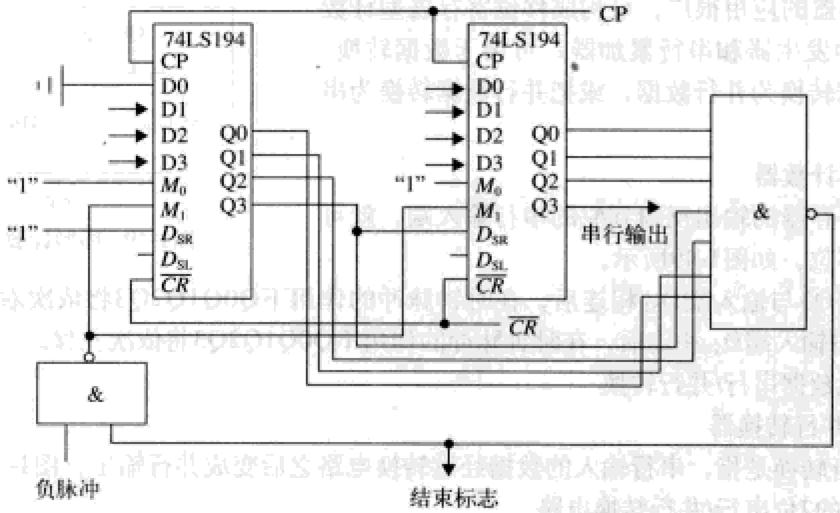


图1-51　7位并行/串行转换电路

四、实验内容及实验步骤

1. 测试74LS194（或CC40194）的逻辑功能

参考图1-48连线，$\overline{CR}$、M_1、M_0、D_{SL}、D_{SR}、D0、D1、D2、D3分别接至逻辑开关的输出插孔，Q0、Q1、Q2、Q3分别接至逻辑电平显示输入插孔。CP接单次脉冲源。自拟表格，逐项进行测试，并与实验指导书给出的功能表做对比。

2. 环形计数器

自拟实验线路，用并行送数法预置计数器为某二进制代码（如0100），然后进行右移循环，观察寄存器输出端状态的变化。再进行循环左移，观察寄存器输出端状态的变化。将结果记录下来。

3. 实现数据的串行/并行转换

按图1-50连线，进行右移串入、并出实验，串入数据自定，自拟表格并记录下实验结果。

4. 实现数据的并行/串行转换

按图1-51连线，进行右移并入、串出实验，并入数据自定，自拟表格并记录下实验结果。

五、实验预习要求

(1) 复习有关寄存器的章节内容，弄懂移位寄存器工作的基本原理。
(2) 查阅74LS194（或CC40194）的资料，熟悉其逻辑功能及引脚排列。
(3) 画好实验要用的表格。

六、实验报告要求

(1) 若要进行循环左移，图1-50和图1-51接线应如何修改？
(2) 画出用两片CC40194构成的7位左移串行/并行转换器电路。
(3) 画出用两片CC40194构成的7位左移并行/串行转换器电路。
(4) 分析实现数据串行/并行转换器、并行/串行转换器电路所得结果的正确性。

实验十三 计数器及其应用

一、实验目的

（1）学会用集成电路构成计数器的方法。

（2）掌握中规模集成计数器的使用及功能测试方法。

（3）运用集成计数器构成1/*N*分频器。

二、实验原理

计数器是数字系统中用的较多的基本逻辑器件，它的基本功能是统计时钟脉冲的个数，即实现计数操作，它也可用于分频、定时、产生节拍脉冲和脉冲序列等。例如，计算机中的时序发生器、分频器和指令计数器等都要使用计数器。

计数器的种类很多。按构成计数器中的各触发器是否使用一个时钟脉冲源来分，可分为同步计数器和异步计数器；按进位体制的不同，可分为二进制计数器、十进制计数器和任意进制计数器；按计数过程中数字增减趋势的不同，可分为加法计数器、减法计数器和可逆计数器。

1. 用D触发器构成异步二进制加法/减法计数器

图1-52是由3个上升沿触发的D触发器组成的3位二进制异步加法计数器。图中各个触发器的反相输出端与该触发器的D输入端相连，就把D触发器转换成计数型T触发器。

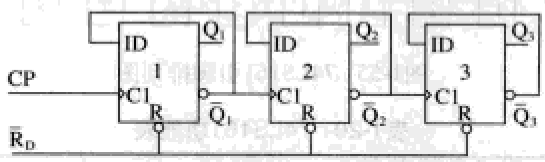


图1-52 3位二进制异步加法计数器

将图1-52加以少许改变后，即将低位触发器的Q端与高一位的CP端相连，就得到3位二进制异步减法计数器，如图1-53所示。

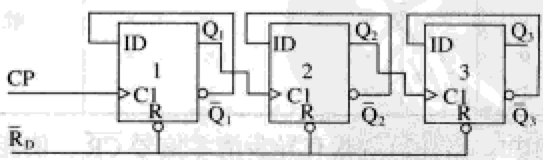


图1-53 3位二进制异步减法计数器

2. 中规模十进制计数器74LS192（或CC40192）

74LS192是同步十进制可逆计数器，它具有双时钟输入，并具有清除和置数等功能，其引脚排列如图1-54所示。

图中，$\overline{LD}$为异步置数控制端（低电平有效），CP_U为加计数脉冲，CP_D为减计数脉冲，$\overline{CO}$为进位输出端，$\overline{BO}$为借位输出端，D_0、D_1、D_2、D_3为计数器预置数输入端，CR为异步清零端（高电平有效），Q_0、Q_1、Q_2、Q_3为计数器数据输出端。

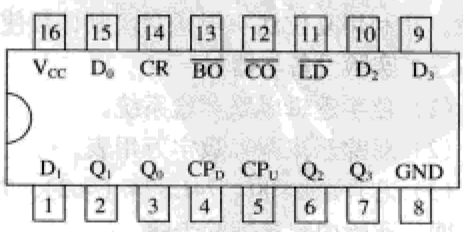


图1-54 74LS192的引脚排列

其功能如表1-19所示。

表1-19　74LS192的功能表

清零	预置	时钟		预置输入				输出			
CR	$\overline{LD}$	CP_U	CP_D	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	d_0	d_1	d_2	d_3	d_0	d_1	d_2	d_3
0	1	↑	1	×	×	×	×	加计数			
0	1	1	↑	×	×	×	×	减计数			
0	1	1	0	×	×	×	×	保持			

3. 4位二进制同步计数器74LS161

该计数器能同步并行预置数据，具有异步清零、计数和保持功能，有进位输出端，可以串接计数器使用。它的引脚排列如图1-55所示，功能表如表1-20所示。

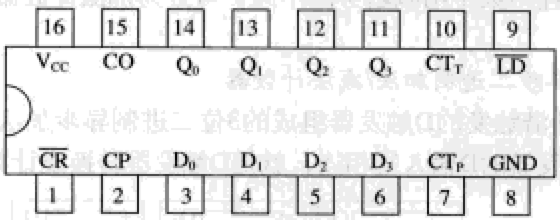


图1-55　74LS161引脚排列图

表1-20　74LS161功能表

清零	预置	使能		时钟	预置输入				输出			
$\overline{CR}$	$\overline{LD}$	CT_T	CT_P	CP	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3
0	×	×	×	×	×	×	×	×	0	0	0	0
1	0	×	×	↑	d_0	d_1	d_2	d_3	d_0	d_1	d_2	d_3
1	1	0	×	×	×	×	×	×	触发器保持，CO=0			
1	1	1	0	×	×	×	×	×	保持			
1	1	1	1	↑	×	×	×	×	计数			

从引脚图和功能表可知，该计数器具有异步清零信号 $\overline{CR}$、使能控制信号CT_T和CT_P、置数信号 $\overline{LD}$、时钟信号CP和四个数据输入端D_0~D_3、四个数据输出端Q_0~Q_3，以及进位输出CO，且$CO=Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot CT_T$。

4. 8421BCD码十进制同步计数器74LS162

74LS162与74LS161的引脚图相同，功能类似。区别在于74LS162为同步清零，十进制计数。在利用复位法实现任意进制计数使用时，注意与74LS161实现电路的不同。

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 双踪示波器和数字万用表。

(3) 芯片74LS00、74LS74（×2）、74LS161（×2）和74LS162（×2）。

四、实验内容及实验步骤

以下实验均在数字逻辑电路实验箱IC插座模块上进行，具体的芯片插法与前述实验相同，

区别在于芯片的功能、引脚不同，芯片之间的连接方法不同。

1. 用D触发器构成3位二进制异步加法计数器

① 按图1-52连线，清零脉冲 $\overline{R_D}$接至逻辑电平开关输出插孔，将低位CP端接单次脉冲源，输出端Q_2、Q_1、Q_0接发光二极管电平显示插孔，各触发器的置位端接高电平“1”。

② 清零后，逐个送入单次脉冲，观察并列表记录Q_2～Q_0的状态。

③ 将单次脉冲改为1Hz的连续脉冲，观察并列表记录Q_2～Q_0的状态。

④ 将1Hz的连续脉冲改为1kHz的连续脉冲，用示波器观察CP、Q_2、Q_1、Q_0端的波形，描绘之。

2. 用D触发器构成3位二进制异步减法计数器

实验方法及步骤同实验内容1，记录实验结果。

3. 测试74LS161的逻辑功能

计数脉冲由单次脉冲源提供，清除端、置数端、数据输入端D_3、D_2、D_1、D_0分别接至逻辑电平输出插孔，Q_3、Q_2、Q_1、Q_0和同步进位输出端接逻辑电平显示插孔。按表1-20逐项测试并判断该集成块的功能是否正常。

4. 测试74LS162的逻辑功能

具体的测试方法同实验内容3，比较与74LS161功能的不同之处。

5. 用复位法构成模7计数器

用1片74LS162（或74LS161）和1片74LS00采用复位法构成一个模7计数器。用单脉冲做计数时钟，观测计数状态，并记录。用连续脉冲做计数时钟，观测并记录Q_D、Q_C、Q_B和Q_A的波形。

6. 用置位法构成模7计数器

用1片74LS162（或74LS161）和1片74LS00采用置位法构成一个模7计数器。用单脉冲做计数时钟，观测计数状态，并记录。用连续脉冲做计数时钟，观测并记录Q_D、Q_C、Q_B和Q_A的波形。

7. 构成模60计数器

用2片74LS162（或74LS161）和1片74LS00构成一个模60计数器。2片74LS162的Q_D、Q_C、Q_B和Q_A分别接两个数码管的D、B、C和A。用单脉冲做计数时钟，观测数码管数字的变化，检验设计和接线是否正确。

五、实验预习要求

(1) 复习计数器的有关原理。

(2) 绘出各实验内容的详细原理图。

(3) 拟出各实验内容所需的测试记录表格。

(4) 查相关资料，给出并熟悉实验所用各集成芯片的引脚排列图。

六、实验报告要求

(1) 画出实验内容中的详细实验原理图。

(2) 记录、整理实验数据及实验所得的有关波形。并对实验结果进行分析。

(3) 总结使用集成计数器的体会。

实验十四 脉冲分配器及其应用

一、实验目的

(1) 学习译码器的使用。

(2) 学习设计和调试较复杂的数字电路。

(3) 学会用示波器测量三个以上波形的时序关系。

二、实验原理

脉冲分配器的作用是产生多路顺序脉冲信号，它可以由计数器和译码器组成，也可以由环形计数器构成，图1-56中CP端上的系列脉冲经N位二进制计数器和相应的译码器，可以转变为2^N路顺序输出脉冲。

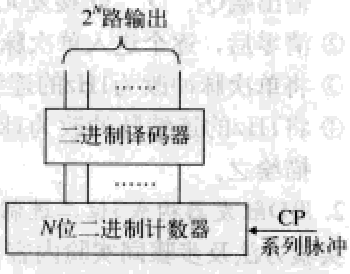


图1-56 脉冲分配器的组成

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 双踪示波器和数字万用表。

(3) 芯片双2-4线译码器74LS139、双JK触发器74LS112（两片）、六反相器74LS04。

四、实验内容及实验步骤

(1) 设计一个用上述器件构成的四相时钟分配器。要求的时序关系如图1-57。

(2) 画出设计逻辑图。

(3) 在实验台上按逻辑图连接线路。用示波器测量CP、A相、B相、C相和D相的时序关系，画出时序图，检查是否满足要求。

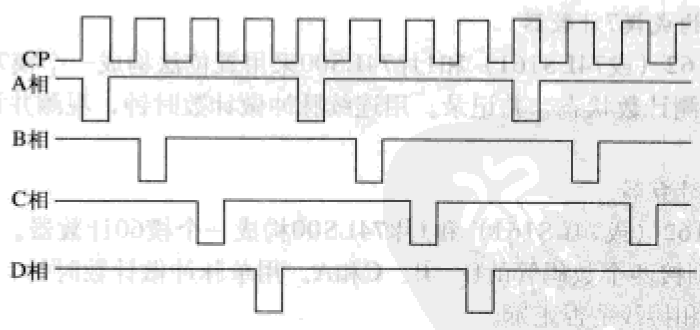


图1-57 四相时钟时序图

五、实验预习要求

(1) 复习有关脉冲分配器的原理。

(2) 按实验任务要求设计实验线路，并拟定实验方案与步骤。

六、实验报告要求

(1) 画出完整的实验原理图。

(2) 写出实验的步骤。写出每一步骤中出现的现象。如果出现错误，则写出解决方法。

(3) 总结分析实验结果，画出下列波形图。

① CP和A相时钟。

② A相时钟和B相时钟。

③ A相时钟和C相时钟。

④ A相时钟和D相时钟。

⑤ CP、A相时钟、B相时钟、C相时钟和D相时钟。

实验十五 时序逻辑电路设计与测试

一、实验目的

（1）掌握时序电路逻辑设计的方法。

（2）熟悉逻辑电路的仿真方法。

（3）学会电路逻辑设计功能检验和调试的技巧。

二、实验原理

设计同步时序逻辑电路时，一般按如下步骤进行。

（1）逻辑抽象得出电路的状态转换图或状态转换表。

要想将要求实现的时序逻辑功能表示为时序逻辑函数，可以用状态转换表的形式，也可以用状态转换图的形式。这就需要如下几步。

① 分析给定的逻辑问题，确定输入变量、输出变量以及电路的状态数。通常都取原因（或条件）作为输入逻辑变量，取结果作为输出逻辑变量。

② 定义输入、输出逻辑状态和每个电路状态的含义，并将电路状态顺序编号。

③ 按题意列出电路的状态转换表或画出电路的状态转换图。

（2）化简状态。

状态化简的目的在于合并等价状态，以求得最简的状态转换图，使设计出的电路有最少的门和触发器。

（3）分配或编码状态。

时序逻辑电路的状态是用触发器状态的不同组合来表示的。首先，需要确定触发器的数目n。因为n个触发器共有2^n种状态组合，所以为获得时序电路所需的M个状态，应取$2^{n-1}<M\leqslant 2^n$。其次，要给每个电路状态规定相应的触发器状态组合，即进行状态编码。在$M<2^n$的情况下，从2^n个状态中取M个状态的组合可以有多种不同的方案，而每个方案中M个状态的排列顺序又有许多种。若编码方案选择得当，设计结果可以很简单。反之，编码方案选的不好，设计出来的电路就会复杂得多。

一般地，为便于记忆和识别，选用的状态编码和它们的排列顺序都遵循一定的规律。

（4）选定触发器的类型，求出电路的状态方程、驱动方程和输出方程。

因为不同逻辑功能的触发器驱动方式不同，所以用不同类型触发器设计出的电路也不一样。为此，在设计具体的电路前必须选定触发器的类型。

根据状态转换图（或状态转换表）和选定的状态编码与触发器的类型，就可以写出电路的状态方程、驱动方程和输出方程了。

（5）根据得到的方程式画出逻辑图。

（6）检查设计的电路能否自启动。

若电路不能自启动，则需采取措施加以解决。

至此，逻辑设计工作已基本完成，图1-58表示了上述设计工作的大致过程。

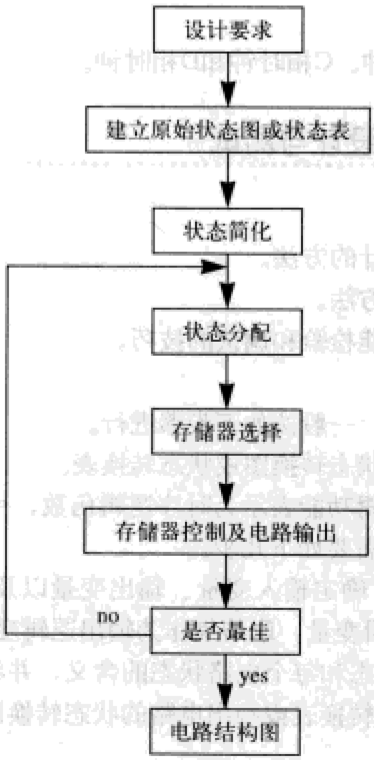


图1-58 时序电路设计过程

由以上设计步骤可见，设计一个时序逻辑电路的步骤较多，过程较繁。若其中一个过程出错，就会导致后面的步骤跟着错下去，最终不能实现功能。要进行修改，几乎就要重新设计，很难临时调整成功。因而，在进行实验调试之前，采用电子设计自动化（EDA）软件进行设计的仿真和元器件与参数的调整，直至满足设计要求后，再进入实验室实测，可大大提高电子设计和实验的效率。

目前流行的各种EDA软件，均可进行设计的仿真与调整。建议读者选用一种自己熟悉的EDA软件，对设计进行仿真后，再进行实验测试。

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 双踪示波器和数字万用表。

(3) 芯片74LS74（两片）、74LS08、74LS00（两片）和74LS161。

四、实验内容及实验步骤

(1) 试设计一个数字二进制码1011序列检测器，即输入序列中连续4个码元符合1011时，检测器输出1。要求输出指示在检测到该序列后灯才亮，并保持一个时钟周期。

(2) 设计一个具有自启动功能的1011序列信号发生器。与实验内容（1）结合，构成一个串行码发送——检测系统。

(3) 设计一个满足图1-59所示时序要求

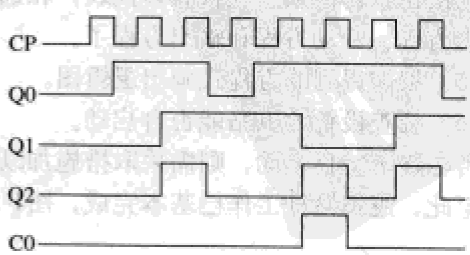


图1-59 时序波形图

的逻辑电路。

五、实验预习要求

（1）掌握时序逻辑电路的一般设计方法。

（2）熟悉EDA软件的使用方法。

（3）根据实验要求设计出实验电路，列出器件清单（使用前序实验已用过的器件），说明功能检验和调试的方法步骤，做出所需记录图表。

（4）用EDA软件对设计的电路进行仿真，绘制（打印）出仿真的波形图。

六、实验报告要求

（1）按实验要求设计出逻辑图，简述工作原理，标明器件型号（附器件清单）。

（2）根据实验结果，做好记录并填入相应的表格，绘制实测的时序图，并与EDA 软件的仿真波形进行比较。

实验十六 EPROM存储器

一、实验目的

（1）熟悉EPROM器件的编程。

（2）学会使用EPROM。

二、实验原理

EPROM是常用的存储器芯片。它具有不易失性，即在无供电电源的条件下，存储的信息也不丢失。在紫外线照射下，能擦除EPROM中存储的数据信息。它主要用于固化程序和存储固定的数据。2716是一个16Kbit（2K×8bit）EPROM芯片。它的引脚图见附录1。其中，A10~A0是地址线，A10是高位，A0低位；$\overline{CE}$为芯片允许端，低电平时允许芯片工作；$\overline{OE}$是输出允许端，低电平时读出数据；O7~O0是数据线，O0是低位，O7是高位；PGM是编程控制端，用于将新的内容写入2716。当$\overline{CE}$是低电平时，在$\overline{OE}$上加低电平，则将A10~A0指定的单元中的8bit数读出，放在数据线O7~O0上。当$\overline{OE}$或者$\overline{CE}$是高电平时，数据线O7~O0处于三态。图1-60是2716读周期时序图。

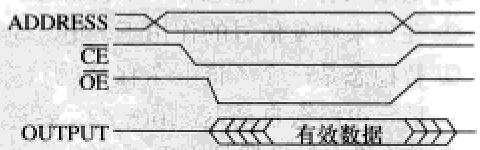


图1-60 2716读周期时序图

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）双踪示波器和数字万用表。

（3）芯片EPROM2716。

四、实验内容及实验步骤

（1）使用DOS下的DEBUG程序，生成一个2KB数据文件，数据内容是00H，01H，02H，…，0FFH。重复多次，一直到2K字节。

（2）使用TDS数字电路实验系统提供的EPROM编程器（或者其他编程器），将文件中包含的2KB的数据写入到2716中。

（3）将2716插到TDS实验台上，接好电源线和地线。2716的地址线A10~A1接电平开关输出，数据线O7~O0接电平指示灯，CS接低电平。

（4）将2716的地址A10~A0置为000H，$\overline{OE}$接高电平。观测并记录O7~O0的值。

（5）将2716的$\overline{OE}$引脚接低电平。从000H开始给A10~A0置值，每次加1，直到03FH。观测并记录O7~O0的值。

（6）将2716的地址A10~A0置为10个不同的随机值。观测并记录O7~O0的值。

五、实验预习要求

（1）复习有关EPROM存储器的原理。

（2）按实验任务要求，设计实验数据，并拟定实验方案与步骤。

六、实验报告要求

（1）写出EPROM2716的写入（编程）步骤。

（2）写出实验内容（4）观测到的O7~O0的值，并予以说明。

（3）写出实验内容（5）中使用的地址及对应的O7~O0的值，并予以说明。

（4）写出实验内容（6）中使用的地址及对应的O7~O0的值，并予以说明。

实验十七　单稳态触发器与施密特触发器

一、实验目的

（1）掌握门电路组成单稳态触发器的方法。

（2）熟悉数字单稳态触发器的逻辑功能及使用方法。

（3）熟悉数字集成施密特触发器的性能及功能。

二、实验原理

1. 单稳态触发器

单稳态触发器具有以下特点。

① 电路只有一个稳态和一个暂稳态。

② 在外来触发信号的作用下，电路由稳态翻转到暂稳态。

③ 暂稳态是一个不能长久保持的状态，由于电路中RC延时环节的作用，经过一段时间后，电路会自动返回到原态。暂稳态的持续时间取决于RC电路的参数值。

由于单稳态触发器具有以上特点，它被广泛地应用于脉冲波形的变换与延时中。单稳态电路有微分型与积分型两大类，这两类触发器对触发脉冲的极性与宽度有不同的要求。

（1）微分型单稳态触发器

它的两个逻辑门是由RC耦合的，而RC电路为微分电路的形式，故称为微分型单稳态触发器。它可由与非门或或非门电路构成。这里我们只看由与非门组成的情况，电路如图1-61所示。

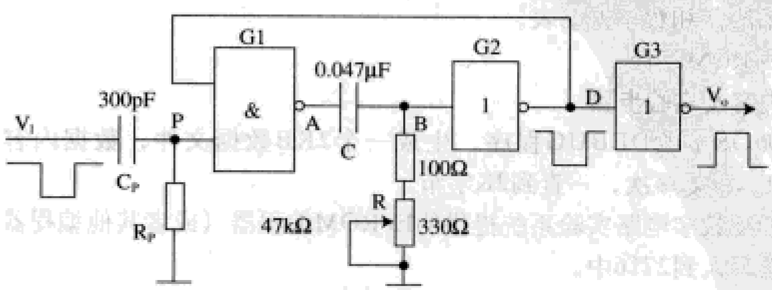


图1-61　微分型单稳态触发器

该电路是负脉冲触发。其中，R_p和C_p构成输入端微分直流电路；R和C构成微分型定时电路，定时元件R、C的取值不同，输出脉宽t_w也不同，$t_w \approx (0.7 \sim 1.3)RC$，与非门G3起整形和倒相的作用。

图1-62为微分型单稳态触发器各点的波形图，一般说来，单稳态触发器有以下几种状态。

① 没有触发信号（$t<t_1$）时，电路处于初始稳态。

② 外加触发信号（$t=t_1$时刻），电路由稳态翻转到暂稳态。

$$V_I\downarrow \rightarrow V_A\uparrow \rightarrow V_B\uparrow \rightarrow V_D\downarrow$$

③ 持续暂稳态一段时间，$t_1<t<t_2$。

④ 当$t=t_2$时，电路由暂稳态自动翻转。

$$C充电 \rightarrow V_B\downarrow \rightarrow V_D\uparrow \rightarrow V_A\downarrow$$

⑤ 恢复过程（$t_2<t<t_3$）。自动翻转时电路不是立即回到初始稳态值，而是要有一段恢复时间。

在$t>t_3$之后，如果V_i再出现负跳变，则电路将重复上述过程。

如果脉冲宽度较小时，则输入端可省去R_p和C_p微分电路。

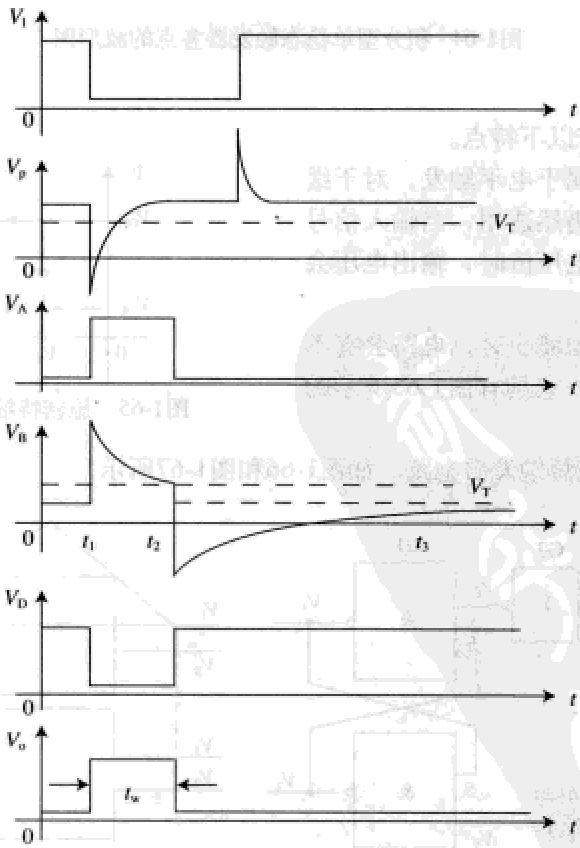


图1-62 微分型单稳态触发器各点的波形图

（2）积分型单稳态触发器

如图1-63所示。

电路采用正脉冲触发，触发脉冲宽度大于输出脉冲宽度的情况，其工作波形如图1-64所示。电路的稳定条件是$R \leqslant 1\text{k}\Omega$，输出脉冲宽度为$t_w \approx 1.1RC$。

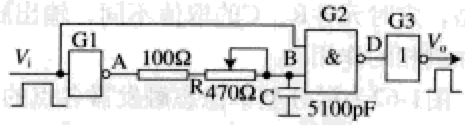


图1-63　积分型单稳态触发器

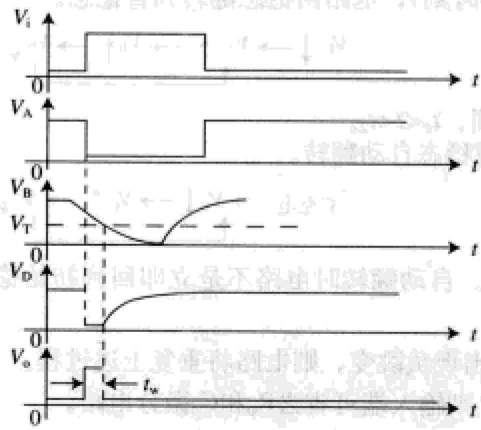


图1-64　积分型单稳态触发器各点的波形图

2. 施密特触发器

施密特触发器具有以下特点。

① 施密特触发器属于电平触发，对于缓慢变化的信号仍然适用，当输入信号达到某一定的电压值时，输出电压会发生突变。

② 输入信号增加和减少时，电路会有不同的阈值电压，它具有图1-65所示的传输特性。

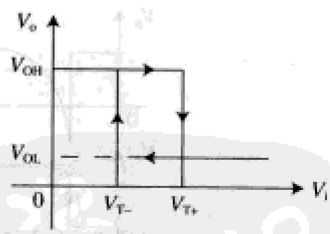


图1-65　施密特触发器的传输特性

有两种典型的施密特触发器电路，如图1-66和图1-67所示。

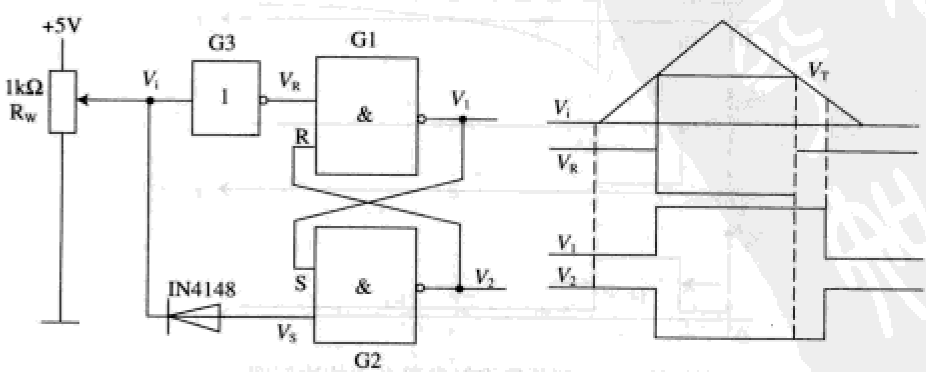


图1-66　由二极管产生回差的电路及各点的波形图

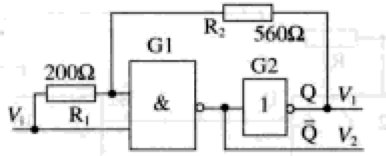


图1-67 由电阻产生回差的电路

这里我们分析一下图1-66所示由二极管产生回差的电路，门G1和门G2是基本RS触发器，门G3是反相器，二极管起电压平移作用以产生回差，其基本工作情况如下。

设V_i=0，G3截止，R=1，S=0，Q=1，电路处于原态。V_i由0V上升到电路的接通电位V_T时，G3导通，R=0，S=1，触发器翻转为Q=0。此后，V_i继续上升，而后下降至V_T时，电路状态不变。当V_i继续下降到小于V_T时，G3由导通变为截止，而$V_S=V_T+V_D$为高电平，因而R=1，S=1，触发器仍保持状态。只有V_i继续下降到使$V_S=V_T$时，电路才翻回到Q=1的原态。电路的回差为$\Delta V=V_D$（V_D为二极管导通电压）。

3. 集成双单稳态触发器CC4528（CC4098）

（1）CC4528的逻辑符号与功能表

图1-68为CC4528的引脚排列与功能表。该单稳态触发器的输出脉宽约为$T_x=R_xC_x$，C_x与R_x为T1和T2端的外接电容和电阻，AI、$\overline{\text{BI}}$为两个正负脉冲触发端，$\overline{\text{CD}}$为复位（放电）端。所有的输出级都有缓冲级，以提供较大的驱动电流。



$\overline{\text{CD}}$	AI	$\overline{\text{BI}}$	Q	$\overline{\text{Q}}$
L	×	×	L	H
H	↑	H	⎍	⎎
H	L	↓	⎍	⎎
H	×	×	保持	

图1-68 CC4528的引脚排列与功能表

（2）应用实例

① 实现脉冲延时，如图1-69所示。

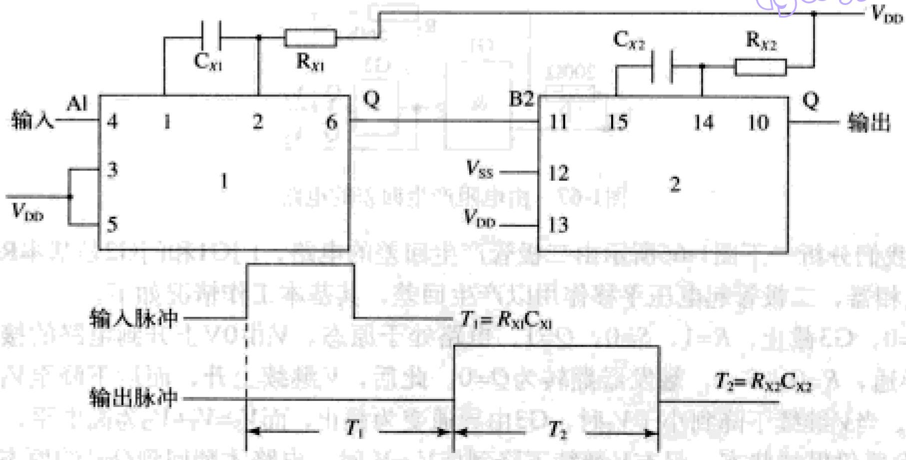


图1-69　实现脉冲延时

② 实现多谐振荡，如图1-70所示。

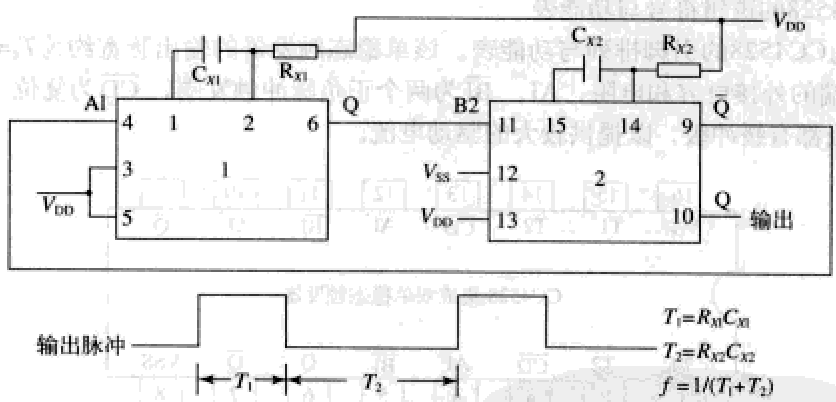


图1-70　实现多谐振荡

4. 集成施密特触发器CC40106

其引脚排列如图1-71。它可用于波形的整形，也可用作反相器或构成单稳态触发器和多

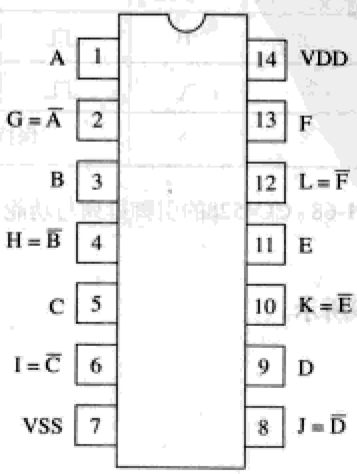


图1-71　CC40106的引脚排列图

谐振荡器。

其应用实例如下。

① 将正弦波转换为方波，如图1-72所示。

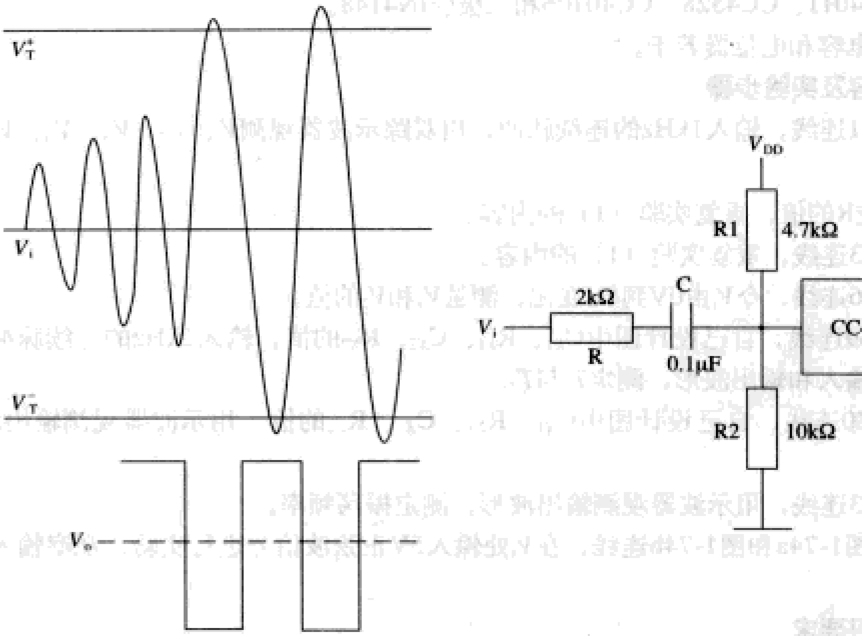


图1-72 将正弦波转换为方波的波形图和电路图

② 构成多谐振荡器，如图1-73所示。

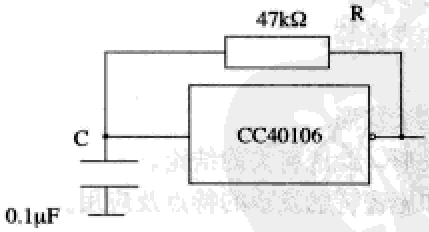


图1-73 多谐振荡器

③ 构成单稳态触发器，如图1-74所示。

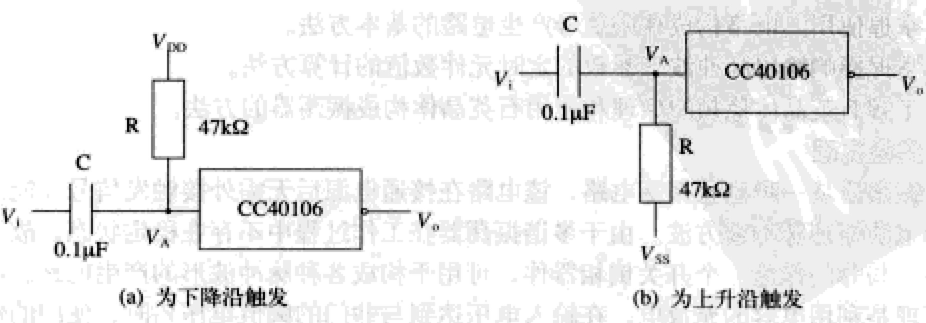


图1-74 单稳态触发器

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）数字万用表、双踪示波器和脉冲源。

（3）芯片CC4011、CC4528、CC40106和二极管IN4148。

（4）电阻、电容和电位器若干。

四、实验内容及实验步骤

（1）按图1-61连线，输入1kHz的连续脉冲，用双踪示波器观测V_i、V_P、V_A、V_B、V_D及V_o的波形，记录之。

（2）改变C或R的值，重复实验（1）的内容。

（3）按图1-63连线，重复实验（1）的内容。

（4）按图1-66连线，令V_i由0V到5V变化，测量V_1和V_2的值。

（5）按图1-69连线，自己设计图中C_{X1}、R_{X1}、C_{X2}、R_{X2}的值，输入1kHz的连续脉冲，用双踪示波器观测输入和输出波形，测定T_1与T_2。

（6）按图1-70连线，自己设计图中C_{X1}、R_{X1}、C_{X2}、R_{X2}的值，用示波器观测输出波形，测定振荡频率。

（7）按图1-73连线，用示波器观测输出波形，测定振荡频率。

（8）分别按图1-74a和图1-74b连线，在V_i处输入5V正弦波信号进行实验，观察输入波形并记录实验结果。

五、实验预习要求

（1）复习有关单稳态触发器与施密特触发器的内容。

（2）画出实验用的详细线路图。

（3）拟定各实验的方法与步骤。

（4）拟好实验用的数据和表格等。

六、实验报告要求

（1）绘出实验线路图，记录波形。

（2）分析各实验结果的波形，验证有关的结论。

（3）总结单稳态触发器和施密特触发器的特点及应用。

实验十八　多谐振荡器

一、实验目的

（1）掌握使用门电路构成脉冲信号产生电路的基本方法。

（2）掌握影响输出脉冲波形参数的定时元件数值的计算方法。

（3）了解石英晶体稳频的原理和使用石英晶体构成振荡器的方法。

二、实验原理

多谐振荡器是一种自激振荡电路，该电路在接通电源后无需外接触发信号就能产生一定频率和幅值的矩形脉冲或方波。由于多谐振荡器在工作过程中不存在稳定状态，故又称为无稳态电路。与非门作为一个开关倒相器件，可用于构成各种脉冲波形的产生电路。电路的基本工作原理是利用电容的充放电，在输入电压达到与非门的阈值电压V_T时，使门的输出状态

发生变化。因此，电路输出的脉冲波形参数直接取决于电路中阻容元件的数值。

1. 非对称型多谐振荡器

如图1-75所示，非门G3用于输出波形整形。

非对称型多谐振荡器的输出波形是不对称的，我们用t_{w1}、t_{w2}和T分别表示充电时间、放电时间和脉冲周期。当用TTL与非门组成时，它们为

$$t_{W1} = RC \qquad t_{W2} = 1.2RC \qquad T = 2.2RC$$

调节R与C的值，可改变输出信号的振荡频率，通常用改变C实现输出频率的粗调，改变电位器R实现输出频率的细调。

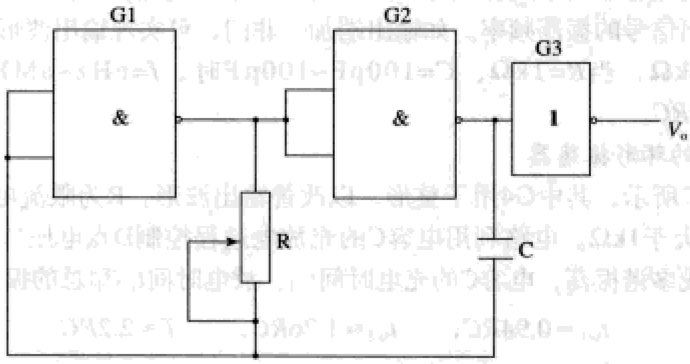


图1-75 非对称型多谐振荡器

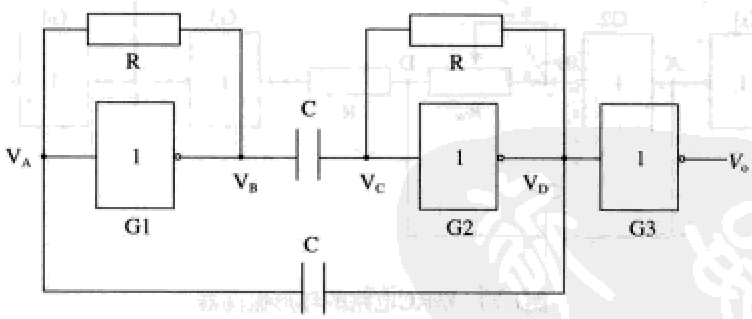
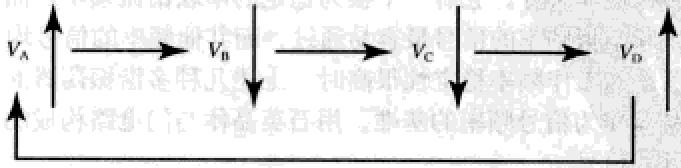


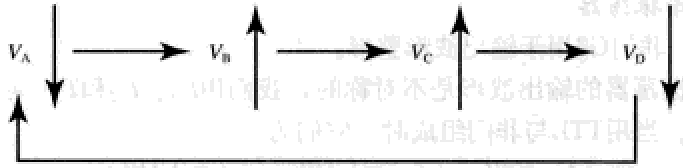
图1-76 对称型多谐振荡器

2. 对称型多谐振荡器

如图1-76所示，设t=0时接通电源，电容尚未充电，此时电路的状态为第一暂稳态。随着时间的增长，电容不断充电，V_A不断增大，直到阈值电压V_T时，电路发生下述正反馈过程。



而后，电容充满电后开始放电，电路又发生下述正反馈过程。



其中，在G1截止且G2导通的瞬间，电路为第二暂稳态。如此，电路将不停地在两个暂稳态之间往复振荡。

由于电路完全对称，电容器的充放电时间常数相同，故输出为对称的方波。改变R和C的值，可改变输出信号的振荡频率。如输出端加一非门，可实现输出波形整形。

一般取$R \leqslant 1\text{k}\Omega$，当$R$=1kΩ、$C$=100pF~100μF时，$f$=nHz~nMHz，脉冲宽度$t_{w1}=t_{w2}=0.7RC$，$T=1.4RC$。

3. 带RC电路的环形振荡器

电路如图1-77所示。其中G4用于整形，以改善输出波形，R为限流电阻，一般取100Ω，电位器R_w要求不大于1kΩ。电路利用电容C的充放电过程控制D点电压V_D，从而控制与非门的自动启闭，形成多谐振荡，电容C的充电时间t_{w1}、放电时间t_{w2}和总的振荡周期T分别为

$$t_{w1} \approx 0.94RC, \qquad t_{w2} \approx 1.26RC, \qquad T \approx 2.2RC$$

调节R和C的值，可改变输出信号的振荡频率。

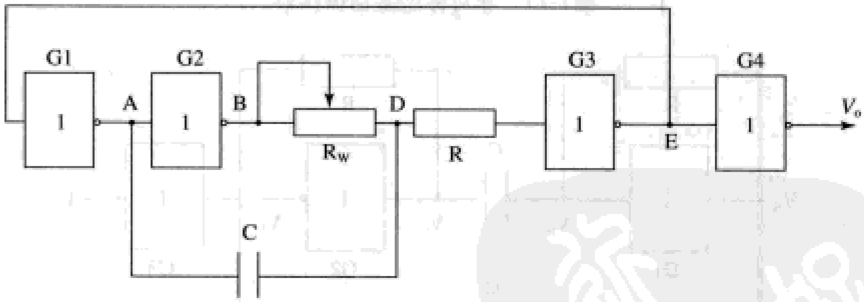


图1-77 带RC电路的环形振荡器

以上这些电路的状态转换都发生在与非门输入电平达到门的阈值电平V_T的时刻。在V_T附近电容器的充放电速度已经很缓慢，而且V_T本身不够稳定，易受温度、电源电压变化等因素以及干扰的影响。因此，电路输出频率的稳定性较差。

4. 石英晶体振荡器

石英晶体的选频特性非常好，它有一个极为稳定的串联谐振频率，而且等效品质因数很高。只有频率等于串联谐振频率的信号最容易通过，而其他频率的信号均会被晶体衰减。

当要求多谐振荡器的工作频率稳定性很高时，上述几种多谐振荡器的精度已不能满足要求。为此常用石英晶体作为信号频率的基准。用石英晶体与门电路构成的多谐振荡器常用来为微型计算机提供时钟信号。

图1-78是常用的晶体稳频多谐振荡器。

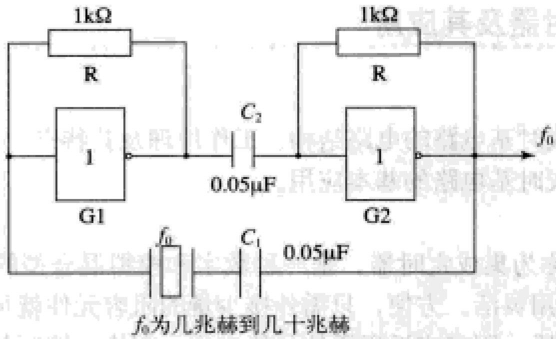


图1-78 常用的晶体稳频多谐振荡器

三、实验设备与器材

（1）数字逻辑电路实验系统。

（2）双踪示波器、频率计、数字万用表和脉冲源。

（3）芯片74LS00（或CC4011）和晶体振荡器4MHz。

（4）电位器、电阻和电容若干。

四、实验内容及实验步骤

（1）用与非门74LS00按图1-75构成多谐振荡器，其中电阻R为10kΩ的电位器，电容C为0.01μF。

① 用示波器观察输出波形及电容C两端的电压波形，列表记录之。

② 调节电位器观察输出波形的变化，测出上限频率和下限频率。

③ 用一只100μF的电容器跨接在74LS00的第14脚与第7脚的最近处，观察输出波形的变化及电源上纹波信号的变化，记录之。

（2）用74LS00按图1-76接线，取R=1kΩ，C=0.047μF，用示波器观察输出波形，记录之。

（3）用74LS00按图1-77接线，其中取限流电阻R为510Ω电阻和R_w取1kΩ的电位器，将两电阻串联，取C=0.1μF。

① R_w调到最大时，观察并记录A、B、D、E及V_o各点电压的波形，测出V_o的周期T和V_o负脉冲宽度值（电容C的充电时间）并与理论计算值比较。

② 改变R_w值，观察输出信号V_o波形的变化情况。

（4）按图1-78接线，晶体振荡器选用4MHz（或其他）的，非门选用74LS00或74LS04，用示波器观测输出波形，用频率计测量输出信号频率，记录之。

五、实验预习要求

（1）复习自激多谐振荡器的工作原理。

（2）画出实验用的详细电路图。

（3）拟好记录实验数据的表格。

六、实验报告要求

（1）画出实验电路，整理实验数据，并与理论值进行比较。

（2）画出实验观测到的工作波形图，对实验结果进行分析。

实验十九 555定时器及其应用

一、实验目的

（1）熟悉555型集成时基电路的电路结构、工作原理及其特点。

（2）掌握555型集成时基电路的基本应用。

二、实验原理

555集成时基电路称为集成定时器，是一种数字和模拟混合型的中规模集成电路，其应用十分广泛。该电路使用灵活、方便，只需外接少量的阻容元件就可以构成单稳触发器、多谐触发器和施密特触发器，因而广泛用于信号的产生、变换、控制与检测。它的内部电压标准使用了3个5kΩ的电阻，故取名555电路。其电路类型有双极型和CMOS型两大类，两者的工作原理和结构相似。几乎所有的双极型产品型号最后的3位数码都是555或556；所有的CMOS产品型号最后4位数码都是7555或7556，两者的逻辑功能和引脚排列完全相同，易于互换。555和7555是单定时器，556和7556是双定时器。双极型的电压是+5～+15V，最大负载电流可达200mA，CMOS型的电源电压是+3～+18V,最大负载电流在4mA以下。

1. 555电路的工作原理

555电路的内部电路方框图如图1-79所示。它含有两个电压比较器、一个基本RS触发器和一个放电开关Td，比较器的参考电压由3只5kΩ的电阻器构成分压，它们分别使低电平比较器V_{r1}反相输入端和高电平比较器V_{r2}的同相输入端的参考电平为2/3V_{CC}和1/3V_{CC}。V_{r1}和V_{r2}的输出端控制RS触发器状态和放电管开关状态。当输入信号输入并超过2/3V_{CC}时，触发器复位，555的输出端3脚输出低电平，同时放电，开关管导通。当输入信号自2脚输入并低于1/3V_{CC}时，触发器置位，555的3脚输出高电平，同时充电，开关管截止。

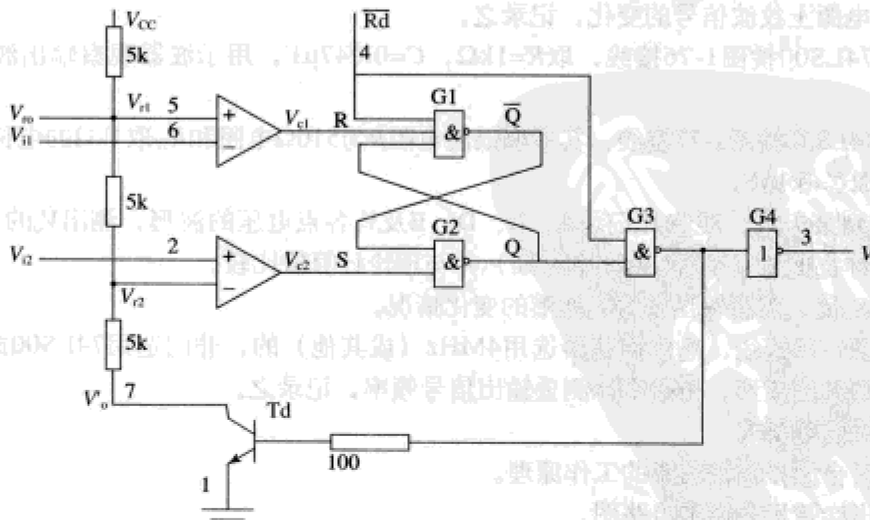


图1-79 555定时器的内部电路图

$\overline{R}_D$是异步置零端，当其为0时，555输出低电平。平时该端开路或接V_{CC}。V_{ro}是控制电压端（第5脚），平时输出2/3V_{CC}作为比较器V_{r1}的参考电平，当第5脚外接一个输入电压，即改变了比较器的参考电平，从而实现对输出的另一种控制，在不接外加电压时，通常接一个0.01μF的电容器到地，起滤波作用，以消除外来的干扰，以确保参考电平的稳定。Td为放电

管，当Td导通时，将给接于第7脚的电容器提供低阻放电电路。

2. 555定时器的典型应用

(1) 构成单稳态触发器

图1-80为由555定时器和外接定时元件R和C构成的单稳态触发器。D为钳位二极管，稳态时555电路输入端处于电源电平，内部放电开关管T导通，输出端V_o输出低电平，当有一个外部负脉冲触发信号加到V_i端。并使2脚电位瞬时低于$1/3V_{CC}$，单稳态电路即开始一个稳态过程，电容C开始充电，V_c按指数规律增长。当V_c充电到$2/3V_{CC}$时，输出V_o从高电平返回低电平，放电开关管Td重新导通，电容C上的电荷很快经放电开关管放电，暂态结束，恢复稳定，为下个触发脉冲的来到做好准备。波形图见图1-81。

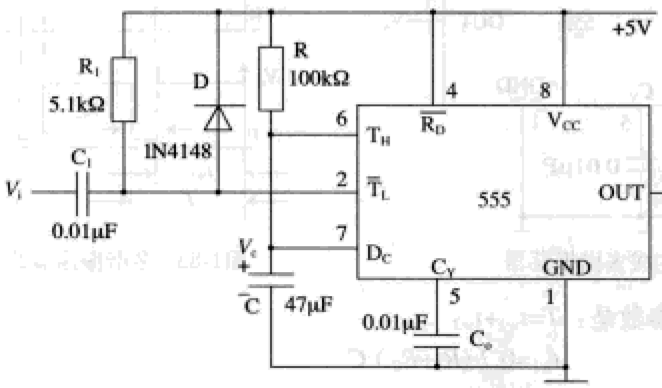


图1-80 555构成单稳态触发器

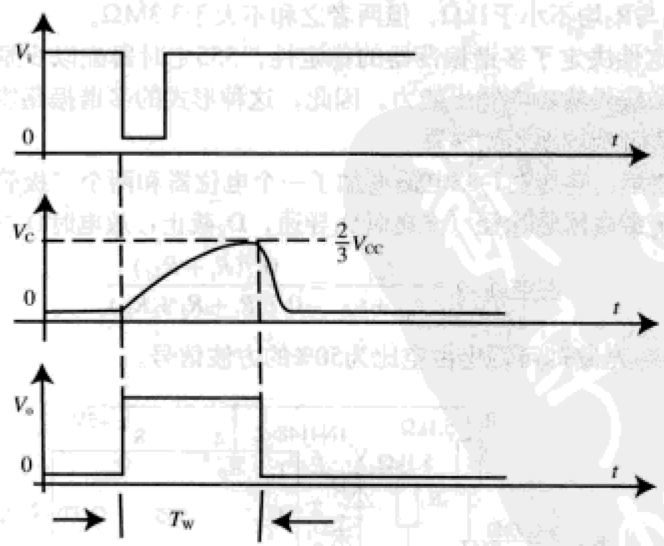


图1-81 单稳态触发器波形图

暂稳态的持续时间T_w（即为延时时间）决定于外接元件R和C的大小，即$T_w=1.1RC$。

通过改变R和C的大小，可使延时时间在几个微秒和几十分钟之间变化。当这种单稳态电路作为计时器时，可直接驱动小型继电器，并可采用复位端接地的方法来终止暂态，重新

计时。

（2）构成多谐振荡器

如图1-82所示，由555定时器与外接元件R₁、R₂和C构成多谐振荡器，2脚与6脚直接相连。电路没有稳态，仅存在两个暂稳态，电路亦不需要外接触发信号，利用电源通过R_1、R_2向C充电，以及C通过R_2向放电端D_C放电，使电路产生振荡。电容C在$2/3V_{CC}$和$1/3V_{CC}$之间充电和放电，从而在输出端得到一系列的矩形波，对应的波形如图1-83所示。

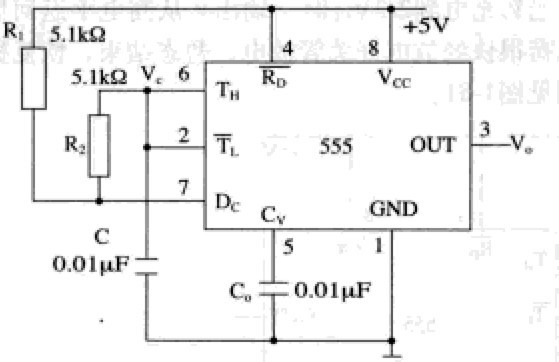


图1-82 555构成多谐振荡器

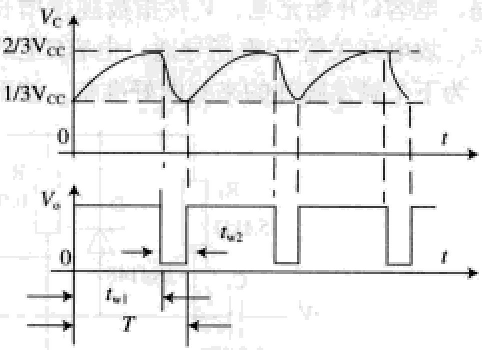


图1-83 多谐振荡器的波形图

输出信号的时间参数是：$T=t_{w1}+t_{w2}$

$t_{w1}=0.7(R_1+R_2)C$

$t_{w2}=0.7R_2C$

其中，t_{w1}为V_c由$1/3V_{CC}$上升到$2/3V_{CC}$所需的时间，t_{w2}为电容C放电所需的时间。

555电路要求R_1与R_2均不小于1kΩ，但两者之和不大于3.3MΩ。

外部元件的稳定性决定了多谐振荡器的稳定性，555定时器配以少量的元件即可获得较高精度的振荡频率和较强的功率输出能力。因此，这种形式的多谐振荡器应用很广。

（3）组成占空比可调的多谐振荡器

电路如图1-84所示，它比图1-82电路增加了一个电位器和两个二极管。D_1、D_2用来决定电容充、放电电流流经电阻的途径（充电时D_1导通，D_2截止；放电时D_2导通，D_1截止）。

$$\text{占空比}\ q=\frac{t_{w1}}{t_{w1}+t_{w2}}=\frac{0.7(R_1+R_{w1})}{0.7(R_1+R_2+R_w)}$$

可见，若取$R_1=R_2$,电路即可输出占空比为50%的方波信号。



图1-84 555组成占空比可调的多谐振荡器

（4）组成占空比连续可调并能调节振荡频率的多谐振荡器

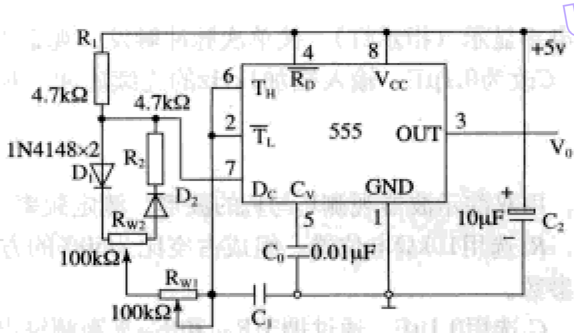


图1-85 555构成占空比、频率均可调的多谐振荡器

电路如图1-85所示，对C_1充电时，充电电流通过R_1、D_1、R_{W2}和R_{W1}，放电时通过R_{W1}、R_{W2}、D_2、R_2。当$R_1=R_2$、R_{W2}调至中心点时，充放电时间基本相等，其占空比约为50%，此时调节R_{W1}仅改变频率，占空比不变。如R_{W2}调至偏离中心点，再调节R_{W1}，不仅振荡频率改变，而且对占空比也有影响。R_{W1}不变，调节R_{W2}，仅改变占空比，对频率无影响。因此，当接通电源后，应首先调节R_{W1}使频率至规定值，再调节R_{W2}，以获得需要的占空比。

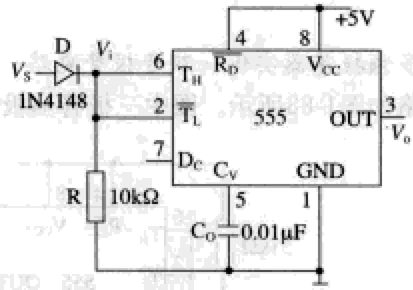


图1-86 555组成施密特触发器

(5) 组成施密特触发器

电路如图1-86所示，只要将2脚和6脚连在一起作为信号输入端，即得到施密特触发器。图1-87画出了V_S、V_i和V_o的波形图。

设被整形变换的电压为正弦波V_S，其正半波通过二极管D同时加到555定时器的2脚和6脚，得到的V_i为半波整流波形。当V_i上升到$2/3V_{CC}$时，V_o从高电平转换为低电平；当V_i下降到$1/3V_{CC}$时，V_o又从低电平转换为高电平。

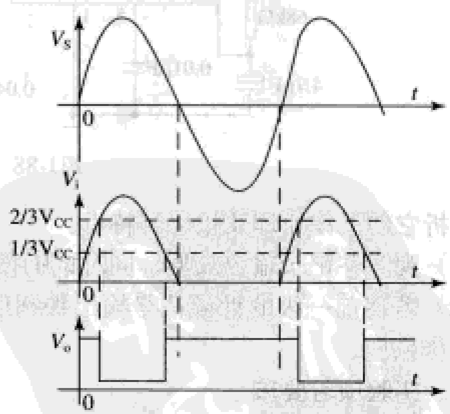


图1-87 555组成施密特触发器的波形图

回差电压为

$$\Delta V=\frac{2}{3}V_{CC}-\frac{1}{3}V_{CC}=\frac{1}{3}V_{CC}$$

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 数字万用表、双踪示波器和频率计。

(3) 芯片NE555、二极管1N4148（2个）、三极管3DG6、扬声器8Ω。

(4) 电阻、电容和电位器若干。

四、实验内容及实验步骤

1. 单稳态触发器

(1) 按图1-80连线，取R=100kΩ，C=47μF，输出接LED电平指示器。输入信号V_i由单次

脉冲源提供，V_0接逻辑电平显示（指示灯）。按单次脉冲触发，观察指示灯的变化。

（2）将R改为1kΩ，C改为0.1μF，输入端加1kHz的连续脉冲，观测V_i、V_c和V_o波形。测定幅度与暂稳态时间。

2. 多谐振荡器

（1）按图1-82接线，用双踪示波器观测V_c与V_o的波形，测定频率。

（2）按图1-84接线，R_w选用10kΩ电位器。组成占空比为50%的方波信号发生器。观测V_c和V_o的波形。测定波形参数。

（3）按图1-85接线，C_1选用0.1μF。通过调节R_{W1}和R_{W2}来观测输出波形。

3. 施密特触发器

按图1-86接线，输入信号的音频信号由正弦信号模拟，预先调好V_i的频率为1kHz，幅度要求稍大于5V（不要过大）。接通电源，观测输出波形，测绘电压传输特性，算出回差电压ΔV。

4. 多频振荡器实例—双音报警电路

电路如图1-88所示。图中三极管基极电阻参数未标，应为10kΩ。

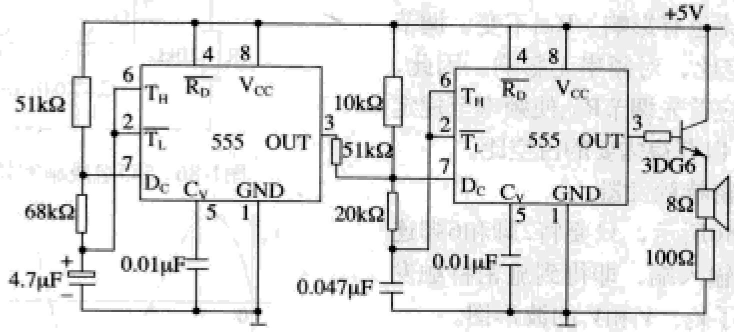


图1-88　双音报警电路

分析它的工作原理及报警声特点。

（1）观察并记录输出波形，同时试听报警声。

（2）若将前一级的低频信号输出加到后一级的控制电压端5，报警声将会如何变化？试分析工作原理。

五、实验预习要求

（1）复习有关555的工作原理及其应用。

（2）拟定实验中所需的数据和波形表格。

（3）拟定各次实验的步骤和方法。

六、实验报告要求

（1）绘出详细的实验线路图，定量绘出观测到的波形。

（2）分析和总结实验结果。

（3）绘出每个谐振电路充放电的等效电路图。

（4）按实验要求选定各电路参数，并按理论计算输出脉冲的宽度和频率。

（5）在双音报警电路中，若将0.047μF的电容分别改为1μF、10μF，对报警声有何影响？

实验二十 D/A与A/D转换器

一、实验目的

（1）熟悉D/A和A/D转换器的工作原理。

（2）学会使用D/A转换器DAC0832和A/D转换器ADC0809。

（3）学会用DAC0832构成阶梯波电压产生器。

二、实验原理

1. D/A转换器

D/A转换器是用来将数字量转换成模拟量的器件。其输入为n位二进制数，输出为模拟电压（或电流）。D/A转换器的形式较多，在集成电路中多采用倒置的R-2R梯形网络。图1-89为一个4位二进制D/A转换器的原理电路。

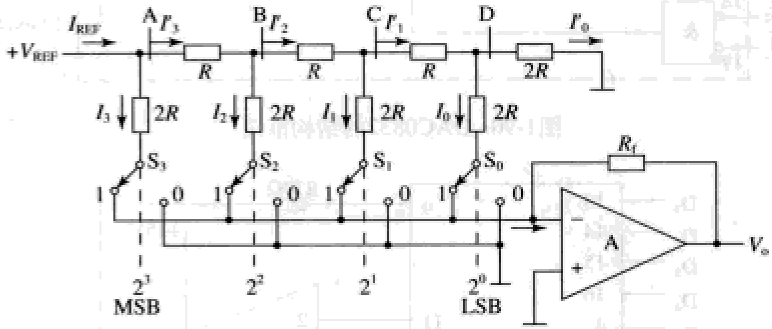


图1-89 D/A转换器的原理电路

它包括由数码控制的双掷开关和由电阻构成的分流网络两部分。输入二进制数的每一位对应一个2R电阻和一个由该位数码控制的开关。为了建立输出电流，在电阻分流网络的输入端接入参考电压V_{REF}。当某位输入码为0时，相应的被控开关接通右边触点，电流I_i（i=0，1，2，3）流入地；输入码为1时，相应的被控开关接通左边触点，电流I_i（$i=0$，1，2，3）流入外接运算放大器。运算放大器的输出电压为

$$v_O=-\frac{V_{REF}R_f}{2^4R}(a_3\times2^3+a_2\times2^2+a_1\times2^1+a_0\times2^0)$$

若将数码推广到n位，可得到输出模拟量与输入数字量之间的一般表达式

$$v_O=-\frac{V_{REF}R_f}{2^nR}(a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0)$$

DAC0832是采用CMOS工艺制成的单片电流型8位D/A转换器，其结构框图如图1-90所示，外引脚图见附录。D_0~D_7为8个数据输入端，当$\overline{CS}=0$，ILE=1，$\overline{WR_1}=0$时，数据送入输入寄存器。当$\overline{XFER}=0$，$\overline{WR_2}=0$时，数据送入转换寄存器。I_{OUT1}、I_{OUT2}为电流输出端，I_{OUT1}可接运算放大器的反相输入端，I_{OUT2}可接运算放大器的同相输入端。Rfb为反馈电阻端，在DAC0832内部有一个15kΩ的电阻，作为运算放大器的反馈电阻。DAC0832采用单电源供电（电压为5～15V），参考电压为–10～+10V，此时，功耗为20mW，转换时间为1μs。

DAC0832的输出形式是电流，外接运算放大器后，可将其转换为电压输出，如图1-91所示。

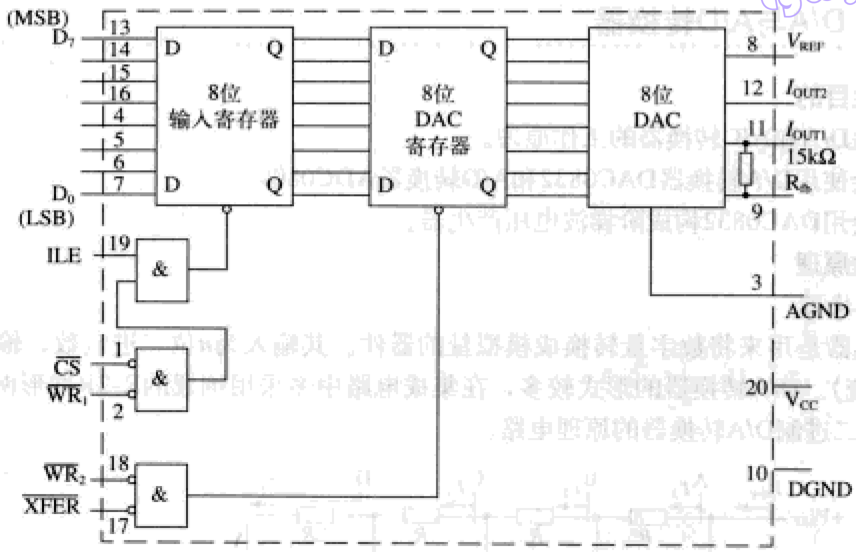


图1-90　DAC0832的结构框图

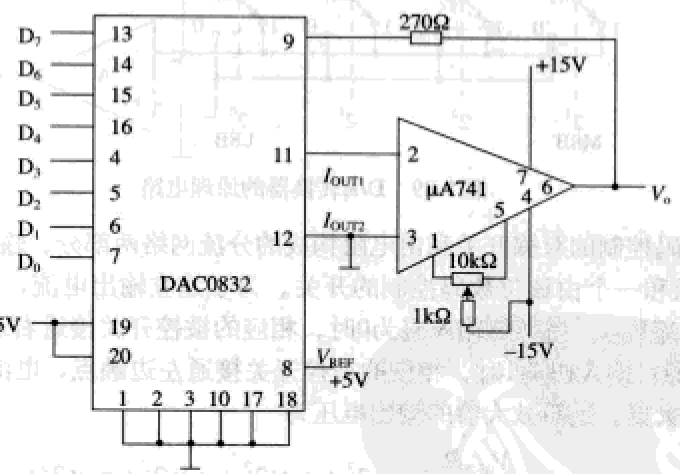


图1-91　DAC0832 电压输出形式

2. A/D转换器

A/D转换器是用来将模拟量转换成数字量的器件。n位A/D转换器输出n位二进制数，它正比于加在输入端的模拟电压。实现模数转换的方法有很多，常用的有并/串型ADC、逐次逼近型ADC和双积分型ADC等。其中，并/串型ADC的速度最快，但成本最高，且精度不易做高。双积分型ADC精度高、抗干扰能力强，但速度太慢，适合转换缓慢变化的信号。逐次逼近型的ADC的特点如下。

（1）完成一次A/D转换所需时间等于（n+2）个时钟周期。n为A/D转换器的位数。

（2）转换精度主要取决于比较器的灵敏度及A/D转换器中的DAC的精度。

（3）输入电压的最大值不仅与A/D转换器的位数有关，而且还与DAC的电路及参考电压有关。

逐次逼近型有转换精度较高、工作速度中等和成本低等优点，因此获得广泛的应用。

ADC0809是采用CMOS工艺制成的单片A／D转换器，其外引脚图见附录1。IN0～IN7为8路模拟信号输入端，可以对0～5V进行转换；ADDC、ADDB和ADDA为地址选择端，当ADDC、ADDB和ADDA均为0时，IN0端被选通；DB0～DB7为8位数字量输出端位。

3. 运算放大器

运算放大器μA741是一种单片高性能内补偿运算放大器，具有较宽的共模电压范围，在使用中不会出现闩锁现象。可用作积分器、求和放大器及普通反馈放大器。

三、实验设备与器材

(1) 数字逻辑电路实验系统。

(2) 数字万用表。

(3) 集成电路DAC0832、ADC0809、μA741、10kΩ电位器，1kΩ电阻和270Ω电阻。

四、实验内容及实验步骤

1. D/A转换器

(1) 在实验箱上按图1-91接好电路。首先通过逻辑开关将D7～D0置成00000000，然后，用万用表测出输出电压。如果输出电压为零，则进入下一步；否则，调节运算放大器的调零电位器，使其输出为0。

(2) 按表1-21中内容依次输入数字量，用数字万用表测出相应的输出电压V_o，将结果填入表1-21中。表中数据采用十六进制。另外，将实测的V_{REF}值填入表1-21中。

(3) 当运算放大器不调零时，即将图1-91中10kΩ短接，并直接接负电源，重测相应的输出电V_o，将结果填入表1-21中。

表1-21 DAC0832数据表

数字量（十六进制）	V_{REF}（V）	V_o（V）	V_o（V）（运算放大器不调零）
FF			
80			
40			
20			
10			
08			
04			
02			
01			
00			

2. A/D转换器

① 在实验箱上按图1-92接好电路。首先将地址选择端ADDC、ADDB和ADDA接地，使IN0端被选通，即模拟信号从IN0端输入。然后，将IN0端接在实验箱上的电压分压电位器上。滑动电位器，可得到0～5V电压。改变电压时，需将START端和ALE端接高电平，而进入A/D转换时，则应将它们接回到低电平。输出端接发光二极管，输出二进制数可通过观察发光二极管得到。

② 按表1-22改变输入电压值，观察输出状态，并将观察到的结果填入表1-22中。

③ 将OE端接低电平，观察电路还能否进行A/D转换。

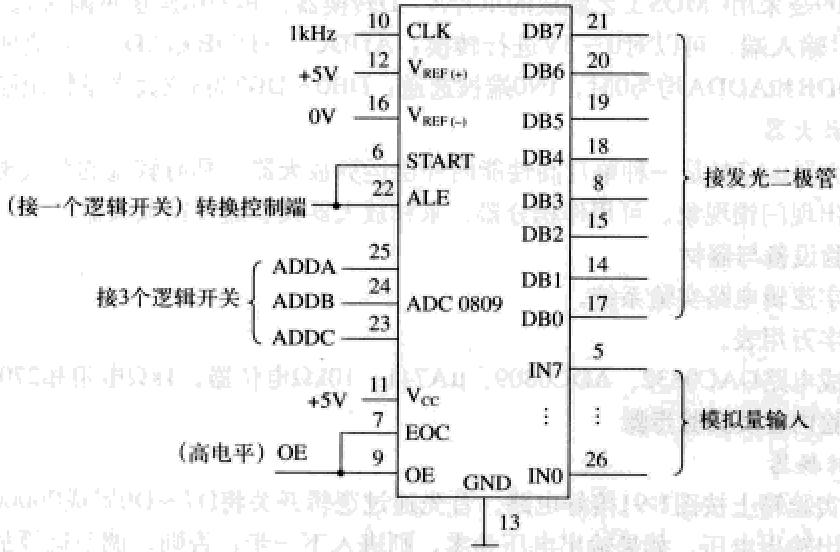


图1-92 A/D转换电路

表1-22 A/D转换数据表

模拟量（V）	0	0.15	0.31	0.625	0.75	1.25	2.5	4.98
数字量（十六进制）								

五、实验预习要求

（1）了解D/A转换器和A/D转换器的功能及其外引脚。

（2）复习D/A和A/D转换器的工作原理。

六、实验报告要求

（1）整理实验数据，并与理论值比较，分析误差原因。

（2）完成以下思考题。

①常用的D/A转换器和A/D转换器的种类有哪些？

②对于8位D/A转换器，当输入二进制数为已知数时，其输出如何计算？

第2章　数字电路与系统课程设计

2.1　专用集成电路设计训练

引言——“自制”集成电路做实验

我们已熟悉用定型的集成电路，如与非门和触发器，连接实验电路进行实验的做法。这里要开始一组新的实验，它的特点是，学生要先将实验电路设计和制作成专用集成电路，再用它进行实验，也就是用“自制”的集成电路做实验。

这种专用集成电路（ASIC，Application Specific Integrated Circuit）技术是半导体技术以及电路CAD/CAM技术发展的产物。将其应用于教学实验，会带来如下好处。

（1）用中小规模集成电路做实验时，需要芯片数较多，外部引线多，连接、检查和排除接触故障费时，实验效率低、用时多，实验电路不能完成太复杂的功能。设计电路时，受芯片结构的约束较多，留给学生自行设计的余地较小。改用ASIC后，由于可将电路集成到一片芯片内，设计时也不必顾虑芯片的选型，因此以上问题均可迎刃而解。

（2）更重要的一点是，ASIC已经得到广泛应用，并在不断发展中。它的设计与制作技术已脱离现行电路连接与实验方法的窠臼，主要采用计算机辅助设计与计算机辅助制作技术。

有多种设计和制作ASIC的方法与手段。其中以可编程逻辑器件（PLD）为“坯料”制作ASIC最简便，因而使用范围也最广。GAL（通用逻辑阵列，Generic Array Logic）是20世纪80年代推出的一种门阵列器件，它的制作采用了EPROM工艺，其内部具有通用型结构，既可实现各种组合逻辑电路，也可实现多种时序逻辑电路，加之器件可重复编程使用，故能满足教学实验的要求。

这里介绍的ASIC技术是数字电路课程设计的基础，它分为两部分。第一部分介绍GAL器件的内部结构和工作原理，介绍设计与编程制作ASIC的软件和编程工具。第二部分为有关设计的实验指导书。为提高效率，第一部分内容围绕设计中所用芯片GAL16V8叙述。

2.1.1　门阵列器件GAL16V8介绍

1. GAL具有逻辑电路的通用结构

从数字逻辑电路的理论可知，数字逻辑电路可分为组合逻辑电路和时序逻辑电路两类，而时序电路又可将其中具有记忆作用的触发器（寄存器）分出，划分为触发器（寄存器）和组合逻辑两部分，如图2-1所示。

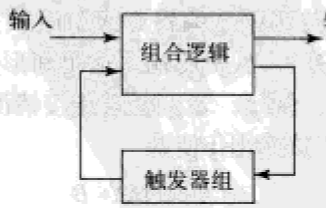


图2-1　时序电路结构

从组合逻辑电路而言，它的输入、输出关系又可归一化为一个或多个与或形式的逻辑表达式。它对应的电路典型结构是一组与门串接一组或门。GAL芯片具有图2-2所示内部结构。

它是由一个与门阵列串接或门阵列和输出单元构成，其中输出单元内含D触发器，将图2-2与图2-1比较可见，GAL内部具有通用的逻辑电路结构，由于输出单元内的D触发器是否接入电路可通过编程控制，因而既可用它实现组合逻辑电路，也可以用它实现时序逻辑电路。

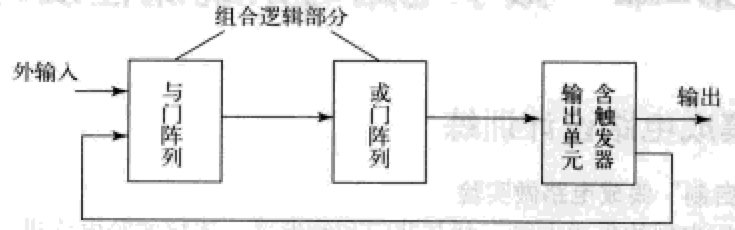
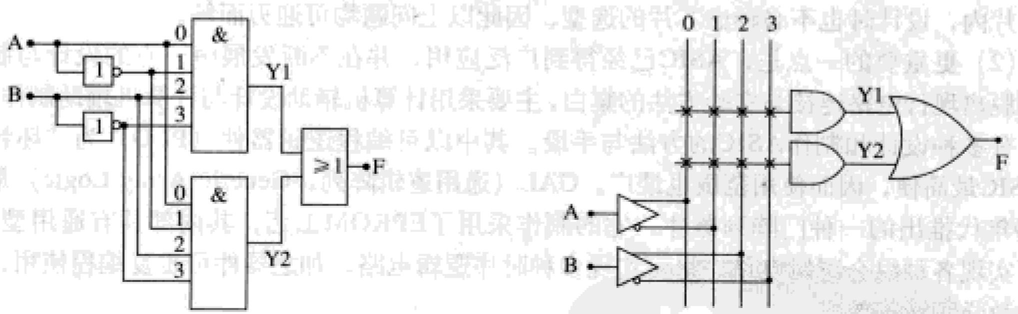


图2-2 GAL芯片内部结构

2. GAL输出通道可以有5种构型

(1) 门阵列芯片内部逻辑结构的表示方法

图2-3a是我们熟悉的逻辑图画法，它表示两个与门Y1和Y2经由一个或门F输出，与门的输入端0、1、2和3分别接外输入A、/A[①]、B和/B。同一结构在讨论门阵列时采用图2-3b的表示方法，图中连线交叉点上的×表示与门一个输入端与一个外输入相连接。通过比较可知，图2-3b的表示方法能较简洁地给出门阵列的内部逻辑结构。



(a) 门阵列结构的逻辑图画法 (b) 门阵列结构的简化画法

图 2-3

从逻辑关系讲，A和/A同时作为与门输入时，其输出恒为0，或门F的输出也是0。如果这种连接格式不能改变，这种内部结构显然没有实用意义。门阵列芯片的特点是，它的内部连接可通过编程方式加以改变（使某些连接断开），从而使输入输出间形成一定的逻辑关系。例如，若编程后连接情况如图2-4所示，或门的输出F便是

$$F=A*/B+/A*B$$

它有2个与项，分别对应与门Y1和Y2。

(2) GAL输出通道的5种构型

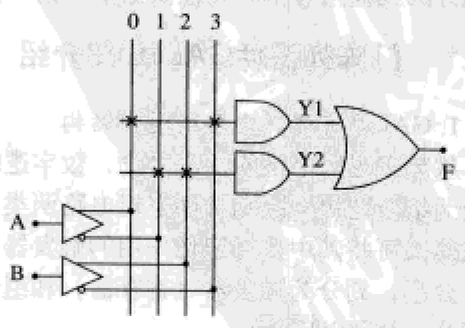


图2-4 编程后连接图

① /A=$\bar{A}$，全书同。——译者注

一片GAL16V8内含8个或门，分别经由8个输出单元形成8条输出通道。GAL的特点是，它不仅可以改变与门输入端与外输入的连接状况，还可以通过写入不同的结构字改变输出与反馈结构，提供下列5种不同的构型。

① I型

I型结构如图2-5所示。它的特点是，三态门控制端接地，或门输出被阻断，外引脚可作输入端使用。

② O型

O型结构如图2-6所示。它的特点是，三态门控制端接高电平，或门输出将立即反映到输出引脚。这是一种典型的组合逻辑电路结构。

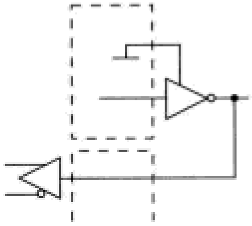


图2-5 GAL输出的I型

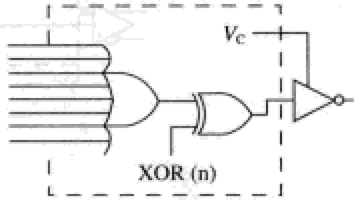


图2-6 GAL输出的O型

③ E型

E型结构如图2-7所示。它的特点是，三态门控制端接一个与门输出端。因此，在进行逻辑设计时，还必须包括设计其三态门控制端的逻辑式。

④ F型

F型结构如图2-8所示。它的特点是，三态门控制端接一个与门输出端。因此，在逻辑设计时，还必须包括设计控制它的逻辑式。此外，它还有一个反馈输入通道，使该输出引脚的信号作为输入变量进入与门阵列。当三态门为高阻态时，该引脚可做输入端使用。或者说，它可作为双向输出端使用。

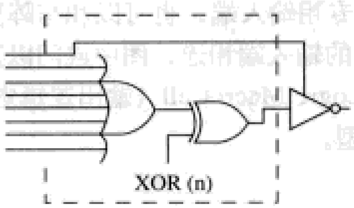


图2-7 GAL输出的E型

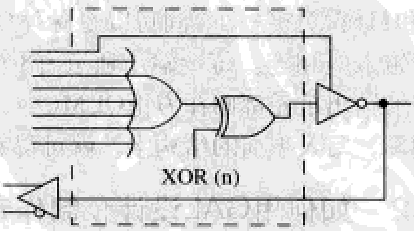


图2-8 GAL输出的F型

⑤ R型

R型结构如图2-9所示。它的特点是，或门输出端与一个内部触发器的D端连接，由该触发器的/Q端引出通向与门阵列的反馈输入，形成该输出触发器的状态信号。注意，它与F型不同，该反馈通道不与外引脚连接，或者说，R型的引脚不能做外输入端使用。此外，各R型的三态门控制端都接在一起，经反相后与外引脚（对GAL16V8而言是11脚）连接。因此，

在逻辑设计时，不必考虑它的三态门控制。R型中的D触发器统一由1脚加入的正脉冲触发，也就是说，选用R型后，1脚将专用于加入触发脉冲，11脚专用于控制三态门。当11脚外加低电平时，三态门导通，输出引脚电平反映内部触发器的状态变化；若11脚接高电平，三态门阻塞，R型通道的外引脚将处于高阻态（此时内部触发器状态仍将正常变化）。1脚和11脚的信号都必须从外部加入。

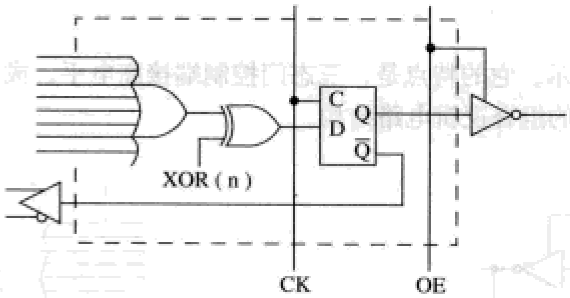


图2-9 GAL输出的R型

（3）高有效和低有效输出

上述O、E、F、R等4种构型可通过写入的结构字控制异或门的XOR（n）端，从而获得高有效或低有效输出，这里举例说明高、低有效概念。我们知道，$F = A*B$和$/F = /A+/B$在逻辑上是等效的。但由于后者的逻辑式左边（输出变量F）有非号，因此，逻辑式右边的值为“1”时，F的取值不同，前者F的值为1，后者F（注意，不是$/F$）的值是0。在进行GAL的逻辑设计时，我们规定引脚的标号本身不带非号，那么，当逻辑式$F = A*B$的右边为1时，F脚上出现高电平，或者说，F脚是高有效输出。若采用逻辑式$/F = /A+/B$，当它的右边为1时，F脚上将出现低电平，F脚是低有效输出。

3. GAL16V8内部结构

图2-10是GAL16V8的逻辑结构图。由图可见，一片GAL16V8有8个通道，图中详细画出了与阵列，其中2脚、3脚、4脚、5脚、6脚、7脚、8脚、9脚和与阵列中的输入线固定连接，1脚和11脚根据结构不同可以是脉冲CK和三态控制OE的专用输入端，也可以和与阵列中的某些输入线相连。每个输出通道有8个与门，它们与或门的输入端相连，图中或门以及结构可变的输出单元都画在方框OLMC中，OLMC 是Output Logic MacroCell（输出逻辑宏单元）的缩写，写入不同的结构字，就可得到上述5种不同的构型。

2.1.2 如何用GAL设计和制作专用集成电路

一片GAL16V8最多可以提供16个输入端（包括输出构型I、F和R所能提供的输入）和最多8个输出端（每个输出通道可接一个D触发器），型号16V8中的16与8即由此而来。

GAL16V8提供了实现多输入和多输出数字电路（组合逻辑或时序逻辑）的可能性。怎样将它制作成我们所需要的集成数字电路呢？这就要按逻辑功能要求改变内部结构，即改变阵列中与门输入端与外输入（包括反馈输入）的连接状况，此外，还要写入合适的结构字使输出构型符合需要。由于与门阵列中的结点有2 048个，结构字也长达82位，要求使用者自行设计结构字和与门阵列中各点通断情况（这些数据统称熔点图，fuse map）显然不

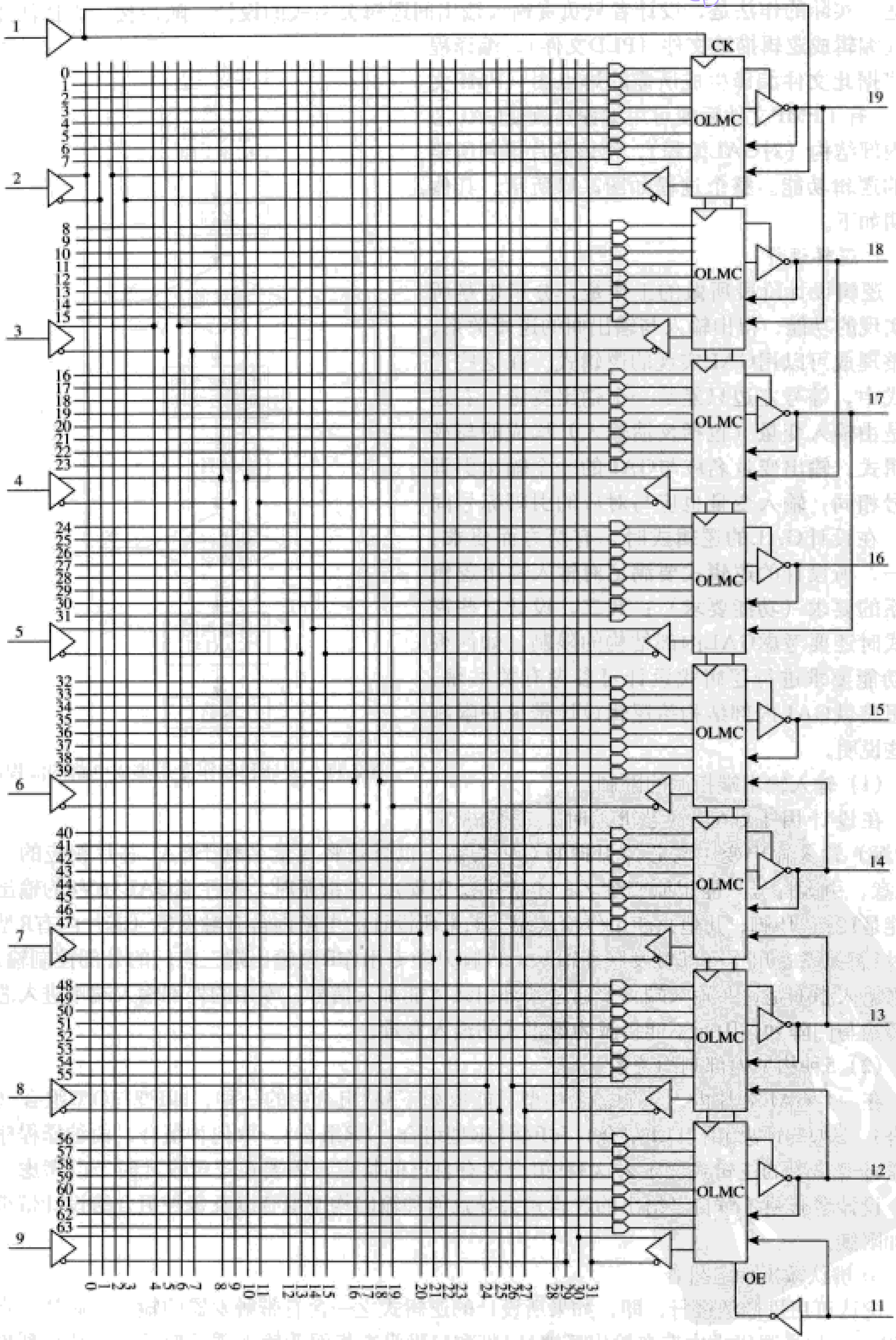


图2-10 GAL16V8的逻辑结构图

合适。实际的作法是，设计者只负责输入输出间逻辑关系式的设计，随后按一定语法要求将其编辑成逻辑描述文件（PLD文件）。编译程序将据此文件编译生成所需的熔点图（FMP文件）。有了FMP文件后便可用编程器改变GAL芯片内部结构（对GAL编程），使该芯片得到所要求的逻辑功能。整个过程如图2-11所示。具体说明如下。

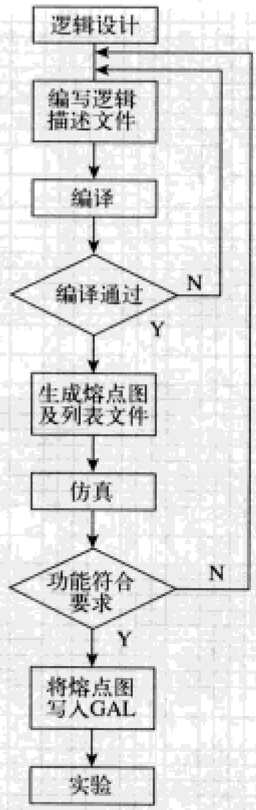


图2-11 设计和制作专用集成电路的流程图

1. 逻辑设计

逻辑设计阶段所做的工作是，分析电路所应实现的功能，得出输入与输出间的逻辑关系，并整理成可以用GAL实现的逻辑式。在这些逻辑式中，等号左边只能是一个输出变量，右边则是由输入变量（包括反馈输入）构成的与或逻辑式。输出变量名应与GAL的一个输出引脚标号相同，输入变量也应与对应的引脚标号同名。在设计GAL的逻辑式时，有两方面要求：其一，所设计的逻辑式要满足对输入输出逻辑关系的要求（功能要求）；其二，设计这些逻辑式时还要考虑GAL内部结构的限制。如何根据功能要求进行逻辑式设计可参考有关书籍，这里将就GAL内部结构给逻辑设计带来的限制做些说明。

（1）输入输出端指派的限制

在设计用于GAL的逻辑式时，先要规定（指派）输入输出变量与芯片引脚的对应关系，也就是将变量名赋予GAL芯片相应的引脚（注意，引脚标号只能用原变量名，不能用反变量）。在指派时，要注意GAL16V8的输出端只能是12脚~19脚，其他只能用作输入端。如果所设计的电路内含有触发器（芯片内有R型），此时1脚只能专用作外部触发脉冲输入，11脚只能专用作R型输出端三态门的外部控制输入。外部输入指的是这些输入端需要通过外部引线才能加入信号，专用的外部输入端不进入芯片内部的与门阵列，因而不能指派为逻辑式的输入变量。

（2）5种构型内部制约关系

在一片GAL芯片内，它的输出构型只能取如下3种组合中的一种，即I型与O型组合（IO组合），E型与F型组合（EF组合）和F型与R型组合（FR组合）。取何种组合，由编译程序根据设计者给出的逻辑式决定。这3种组合各有自己的特点，需要在逻辑设计时予以考虑。因此，设计者必须了解自己给出的逻辑式会导致何种输出构型组合以及每种组合给设计带来的附加限制。

① 辨认输出构型组合

辨认可用排他法进行，即，如果所设计的逻辑式之一含有带触发器的输出，芯片一定是FR组合，如果逻辑式中存在输出反馈且1脚和11脚没有指派为输入逻辑变量，芯片构型也将

是FR组合；如果逻辑式中有输出反馈且1脚或11脚指派为输入逻辑变量，构型将是EF组合；如果不存在上述两种情况，芯片结构就是IO组合。

② 每种构型组合的特殊问题

- IO组合。15脚和16脚不允许指派为输入（I型），因为I型输入是经由相邻通道进入与门阵列的，从图2-10可见，15脚和16脚没有经由相邻通道的反馈通道。因此，它们不可能指派为专用输入端。
- EF组合。每一有效的输出通道都要对它的三态门控制列写控制方程，否则输出引脚对外呈高阻态。12脚和19脚固定为E型，因为EF组合中，这两条通道的反馈输入分别被1脚和11脚占用。
- FR组合。1脚固定用作外加触发脉冲的输入端，11脚固定用作触发器输出通道的三态门控制。对于其中组合逻辑输出通道（F型），一定要分别设计其三态门控制方程，否则它的引脚对外呈高阻态。

③ 每种构型的与门数

每个通道都有8个与门。对于O和R输出，它的输出逻辑式中最多允许有8个与项。对于E和F输出，因为要用一个与门控制其三态门，因而它的输出逻辑式中最多只能含7个与项。

2. 编辑逻辑描述文件（PLD文件）

设计好逻辑式后，即可启动GS程序，在此软件帮助下进行ASIC的设计、编译与仿真流程。

在DOS提示符后键入GS，按回车后即出现屏幕信息，说明GS特点与中文信息使用办法（对于不具图形卡的计算机，再按一次键即可）。按回车后将显示以下菜单（menu）

```
GAL COMPILER and SIMULATOR
*MENU*
Editor
CLPE Compiler
GAL16V8 Simulator
DOS command
Arrow keys: select;  <Enter>key: comfirm; <ESC>key: exit; <?>
```

菜单中各项分别是文本编辑、熔丝图编译、GAL16V8功能模拟和DOS命令。底行提示操作者可用光标键选择菜单项，用（Enter）键确认，用（ESC）键退出菜单返回，用<?>键可显示中文信息。

用光标键选择菜单中的Editor（编辑器）项，按<Enter>（回车）键，按照提示给出文件名后（扩展名通常取 .PLD），即可编写或修改逻辑描述文件。

（1）文本编辑Editor介绍

PLD文件可用一般文本编辑程序生成，但在GS环境内用Editor进行编辑与修改更方便省时。选取Editor后，将出现编辑屏幕，在“Source file name = ”后面，用户输入待编辑的文件名，按回车键后系统进入屏幕编辑状态。在屏幕编辑状态下，可按常规使用如下一些编辑用键，如Del、Enter、Ins、Home、End、Pgup、Pgdn以及4个光标移动键等，组合键<Ctrl-Y>的功能与wordstar相似，用它来删除当前行。考虑到逻辑描述文件大多不超过50行，上述编辑用键已够用了。

编辑完成后，可根据屏幕下方的提示行按<F2>键（Save），屏幕上将显示

"Destination file name = ",

其后自动显示当前的编辑文件名，如不拟修改，按回车键即可。按回车键后，改好的文件将写回磁盘，取代盘内同名的文件，如欲中途退出编辑状态，可按<ESC>键。

<F10> 和 <?> 是HELP键，按它将显示Editor中使用的编辑键功能和PLD文件中所用关键字的英文和中文说明。

（2）PLD文件语法

下面先举一例，该例的文件名为EXAM2.PLD。据此我们讨论在编写PLD文件时需遵循的语法要求。

例：EXAM2.PLD

```
pld16V8
Example of GAL design
This is a stepping motor controller with speed,
direction and stop contorl. It shows that one
GAL chip can replace a moderate complex digital
circuits.
Designed by W. B
Mar. 21, 2002
signature
TEST-1
pindef
CL; clock
SP; speed control
ST; stop control
DI; direction control
OE2; SS output enable
NC NC NC NC GND
OE1; flip flop output enable
NC
SS; intermediate control signal
A; stepping motor control
B;
C;
CL2; second clock
NC NC PWR
equation
/A:=/DI*/SS*B*/C+DI*/SS*/A*B+/DI*/SS*/B*C+/DI*/SS*/A*C
+SS*/A*C+SS*/A*B;
/B:=DI*/SS*C+DI*/SS*/A*/B*/C+/DI*/SS*A+/D1*/SS*/A*/B*/C
+SS*/B+SS*A*B*C;
/C:=DI*/SS*A +DI*/SS*/A*/B*/C+DI*/SS*B+/D1*/SS*/A*/B*/C
+SS*/C+SS*A*B*C;
/SS=/ST*/SP+/ST*/CL2;
/CL2:=CL2;
SS.OE=OE2;
```

```
end
```

现通过此例说明PLD文件的语法要求。

PLD文件按顺序使用如下5个关键字，它们是（为区别起见，上例中关键字均小写，实际编程时不作限制）pld、signature、pindef、equation和end。

① pld。它必须起始于第一行第一列，以pld为前导，随后紧接器件型号16V8（在pld与16V8间不留空格）。其下数行为用户的说明或其他信息，如设计日期、设计者姓名和功能等。

② signature（电签名）。它仅用于GAL，在signature下面一行的字符（不超过7个）将写入GAL作为该芯片的设计信息，如设计日期和型号等。只要芯片不改写，即便芯片已加密，这一信息仍可读出，因此叫电签名。

③ pindef。在此关键字后开始依次定义GAL16V8芯片全部20个引脚的标号。引脚标号前不可加非号，各引脚标号后可加注释（但注释前必须加一个分号）。逻辑式中不用的引脚用NC作为标号。例中引脚标号给定顺序与引脚顺序同。参看图2-12引脚分配图。

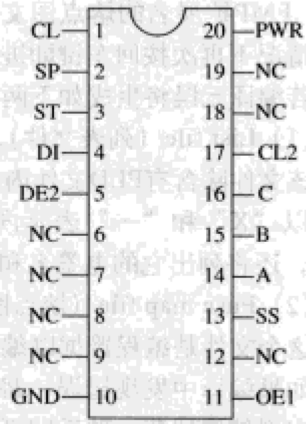


图2-12 GAL16V8引脚分配图

④ equation。在此关键字后开始列写引脚间逻辑式。列写时必须注意如下几点。

- 有效逻辑运算符为 * 、+ 和 / 三种，“*”表示逻辑与，“+”表示逻辑或，“/”表示逻辑非。逻辑式内不能有括号。
- 逻辑式左边的变量必须是器件输出端的标号。输入端的标号不允许出现在逻辑式左边。
- 逻辑式左边的输出量取非值时（在有关的输出端标号前加“/”表示它的逻辑非），说明该输出端是低有效，即逻辑式右边为1时，输出端为低电平。
- 输出端构型为O或R时，等号右边允许有8个与项。输出端构型为E或F时，等号右边只允许有7个与项，另一与项专用作三态门控制。
- 三态门输出控制需要单列逻辑式编程（OE控制）。用如下格式

  ```
  Output.OE=Expression;
  ```

  其中，Output为有关输出端的标号，等号右边的表达式（Expression）中只能含一个与项，不允许有或项。
- 当要求输出端带有触发器时，等式前面要加冒号，即“:=”。
- 每一逻辑式后需加分号（；），分号后可写入少量说明文字。
- 数字1与0也可作为与项。与项为1时，与阵列中有关熔点全部断开，输出恒为高电平，如果要输出三态门保持打开，便可令其OE控制项等于1。与项为0时，与阵列中有关熔点全部保持原状，可用它调整被编程的与项位置。
- 列写逻辑式时，每行字符长度不宜超过80，超过80的逻辑式可以分行列写，在“*”或“+”号前分行均可（见上例）。

⑤ end。此关键字表示逻辑列式终止。PLD文件到此结束。

3. 编译PLD文件

从菜单上选CLPE Compiler项后，按<Enter>将依次出现如下提示

```
Source file name =
List file name =
Fuse map file name =
```

键入源文件名后，按回车键即出现带有 .LST扩展名的列表文件名，再按回车键即出现带有 .FMP扩展名的熔点图文件名。出现文件名后，如觉不妥，可修改或按<ESC>键退出。正常情况下再次按回车键即进入编译。

若编译无误将生成如下两个文件。

(1) List file（列表文件），扩展名为 .LST。

该文件除含有PLD文件内容外，还将以图形方式表示出各引脚的标号和列出内部熔点图。图中以“X”和“—”表示内部逻辑结点的通断状态，“X”表示通，“—”表示断。对于GAL，还将列出它的电签名和部分结构参数。

(2) Fuse map file（熔点图文件），扩展名为 .FMP。

这个文件是编程器据以编程的数据文件。

如果编译中发现错误，将中止编译并显示错误信息，此时按下<F1>键（屏幕上有提示）便将出现编辑屏幕，随后即可进行修改。

• 编译错误信息

Can't find type definition at line 1　（在第一行找不到型号定义）

No such type　（芯片型号有误）

Miss key word <PINDEF>　（缺关键字<PINDEF>）

NO pin definition　（缺引脚定义）

Pin number not match　（所定义的引脚数不对）

Pin name repeated　（引脚名重复，注意，除NC外，不可以有重复的引脚名）

More logic equation than needed　（逻辑式过多）

'=' is needed　（逻辑式中缺“=”号）

' NC 'can't be defined as logic variable　（不能用NC作为逻辑变量）

OE equation repeated　（同一OE控制有多个方程定义）

output equation repeated　（同一输出量有多个方程定义）

Can't be output　（此引脚不能输出）

There isn't OE control　（此处不应有OE控制）

Can't have registered Output　（此处不具有带触发器的输出）

Registered equation is needed　（应为带触发器的输出式）

Can't input　（等式右边有不能作为输入的逻辑变量）

Miss OE item　（缺OE控制）

Extra OE product item exist　（OE控制式中有一个以上的乘积项）

Extra product item exist　（逻辑式中含多余的乘积项）

Active level wrong　（逻辑式中有效电平规定与芯片要求不符）

Miss key word<END>　（缺关键字<END>）

Miss key word<EQUATION>　（缺关键字<EQUATION>）

Miss ';'（逻辑式后缺"；"号）

False input variable---> XX （箭头指出的输入变量XX有误）

Can't use '/' in pin name definition.（在规定引脚名时不能用"/"）

T00 many input variables.（乘积项中变量数超过允许数）

File access failed.（文件存取失败）

4. 单步和多步仿真

编译通过后，只能说明设计的PLD文件语法无错误，且与GAL内部结构没有矛盾，但在逻辑功能上是否满足设计要求尚有待检验。直接的检验方法是将编译所得的熔点图文件写入GAL芯片（用编程器），然后接入电路中加上适当电平检验其逻辑功能。这种方法比较费时，一旦逻辑功能不符合要求还要回到第1步（逻辑设计）重新开始。对于初学者，这种反复可能更多。为了提高实验效率，GS程序中提供了单步和多步仿真功能。选择菜单中的"SIMULATOR"并按回车键后即进入仿真程序。

首先，操作者要装入编译器生成的熔点图文件（FMP文件），此后屏幕上将显示GAL16V8的引脚图，引脚侧显示它的属性（构型）A和电平V。引脚属性有I、O、E（e）、F（f）和R等5种，它们的特点如上所述，其中小写e和f分别表示E与F型通道的OE项等于0。引脚的输入电平用大写表示，有C、L、H和X等4种，它们分别表示触发脉冲（C）、低电平（L）、高电平（H）和不定值（X），输出电平用小写表示，有l、lz、h、hz和z等5种，它们分别表示低电平（l）、高电平（h）、高阻态（z）、lz和hz [分别表示三态门阻断后（此时引脚呈高阻态）触发器的内部输出状态（低电平和高电平）]。

（1）在仿真前应先进行如下设置操作。

① 设置输入电平。用光标移动键将光标移至需要设置电平的输入引脚，按下H、L、C和X等键即可设置相应电平，其中触发脉冲（C）只有当芯片为FR构型时方可设置，且只能设置在1脚处。

② 设置内部触发器状态。用光标移动键将光标移至需要设置状态的R型输出端，按下H和L键即可设置相应的触发状态。触发器状态不可设置为"X"，在装入FMP文件后，内部触发器的初始状态均自动设置为"H"，这是因为GAL芯片在上电时有此预置功能。

③ 设置引脚之间的连接。由于芯片内部设有反馈通道，一般情况下，无需在芯片引脚间外加连线。但在某些情况下，外部连线还是必要的。GS模拟程序提供了如下连线功能。

- 在芯片选用FR构型时，1脚和11脚需外部信号，此种信号如果设计由GAL的某一输出端提供，可通过连线操作实现。
- 在任一输入端，都可令其与外部时钟（clock）相连。与时钟端相连的输入端在每一次仿真操作后，它的电平做相反变化（注意，从屏幕上看，该端显示的是仿真后的输入电平）。芯片为FR构型时，如果令它的专用触发端与外部时钟相连，当电平由低（L）转换为高（H）后，产生一次触发（显示为"C"）。
- 设置和取消连接操作由A和D两键实现。A键为增加新的连接，D键为取消某一连接。
- 按下A键后，屏幕上将出现有关提示，按提示信息键入输出和输入引脚号即可，如

果要将外部钟时与输入相连，输出引脚号应键入“O”。

• 按下D键后，按提示信息键入需要取消连接的输入引脚号即可。

④ 如果电平或连接设置有问题，屏幕上将出现有关提示。

⑤ 按下<F10>键，屏幕上将出现帮助信息。

(2) 电平和连接设置好后，便可进行仿真操作。

GS提供了单步和多步仿真的功能，通过按<F8>键转换。在单步仿真时，屏幕上显示GAL的引脚图，按S键进行一次仿真过程。

多步仿真时，引脚编号（PIN）、标号（LBL）和属性（A）分别占用一列，其余20列顺序显示每一步的电平状态，单步仿真的电平变化信息（20步以内）也都记录并显示在屏幕内。多步仿真时不能设置引脚之间连接，但单步仿真设置的连接转为多步仿真时仍有效。

多步仿真时仿真操作键有3个，即S、I和A。其中 S键和 I键均为单步仿真键，按一下仿真一次。仿真时，所用输入和状态数据为光标所在列的数据。用S键仿真时，当前步的输入电平、输出电平和触发器状态将自动传到下一步；用I键仿真时，仿真得到的状态变化传送到下一步，但输入电平不向后传送。这种仿真方式有利于设计者事先设置好各步的输入电平，然后逐步仿真观察。A键是多步仿真键，按下后GS将自光标所在列向后仿真，所用仿真方式与I键相同，因此要先设置好各步的输入电平。

由于仿真操作从光标所在处开始，因此设计者可通过调整光标位置和适当修改输入电平与初始状态，反复测试电路的输出特性。

多步仿真时，按<F9>键可实现电平显示形式的转换（字符形式—波形图形式）。

在设计异步时序电路时（如后面的交通灯控制实验）要注意，GAL引脚上设置C（外加时钟）时，仿真时触发器状态将发生变化。如果要单独检查输入电平引起组合电路部分的输出，就应将1脚的电平设置为H、L或X。

按下仿真键后，如果需要补充信息，屏幕上将出现有关提示。如果逻辑功能设计正确，且电平设置合理，输出电平应出现所需要的变化。否则就应返回Editor，修改PLD文件，经编译后再次进行仿真试验，直至无误。

• 仿真错误信息

pin xx can’t set external input.（引脚 XX 不能设置外输入）

pin xx can’t use clock input.（引脚 XX 不能设置时钟输入）

pin xx input level should be defined.（引脚 XX 电平应设定）

pin xx register’s status should be defined.（引脚 XX 触发器状态要设定）

pin xx registered 0utput can’t set ‘X’.（引脚 XX 是触发器输出，不能置“X”）

pin xx is not axailable output pin.（引脚 XX 不是有效输出端）

pin xx is not available input pin.（引脚 XX 不是有效输入端）

pin xx can’t link to input in this architecture.（在此构型中，引脚 XX 不能连接输入端）

PLD type isn’t GAL16V8.（所用FMP文件的PLD型号不是GAL16V8）

File acess failed.（文件存取失败——可能文件名不对）

Disk read failed.（读盘失败）

It isn’t FMP file.（这不是FMP文件）

5. 对GAL进行编程

确定FMP文件正确后，即可对GAL16V8进行编程，GAL编程用设备由实验室指定。

6. 实验

GAL16V8编程后，便可在实验箱上连线实验。

实验一 全加器设计训练

一、 实验目的

（1）复习组合电路设计方法并设计组合逻辑电路。

（2）初步掌握GAL性能与设计流程。

（3）学会使用GAL的设计与调试手段：GS软件包和通用编程器。

二、实验原理

组合电路设计过程。组合逻辑电路的特点是它的输出状态只与当前的输入状态有关。它的设计过程大体由如下步骤组成。

（1）需求分析，也就是了解和分析设计要求。

（2）决定所需的输入和输出变量数，并赋以文字符号。

（3）根据分析所得输入输出关系，列出真值表。

（4）将每一输出变量与输入量之间的关系，用逻辑方程表述并加以简化。

（5）画出逻辑电路图。

在设计用GAL实现的电路时，不必画出逻辑图，设计者只需将逻辑式按一定规格写成逻辑描述文件，后面的工作将由计算机和编程器完成。举例说明如下。

例：设计一个全加器，框图如图2-13所示。它的输入为x、y和z，其中x和y分别代表一位二进制数，z代表低位的进位信号，输出为S和C，S代表和值，C代表向高位的进位信号。

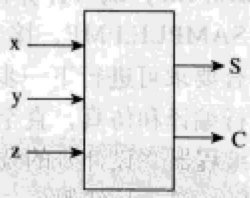


图2-13 全加器框图

根据上述全加器功能，结合对二进制求和过程的理解，可得出表2-1所示真值表。

表2-1 全加器真值表

z	x	y	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

根据真值表，可以直接写出如下逻辑式

$S = /z*/x*y + /z*x*/y + z*/x*/y + z*x*y$；

$C = /z*x*y + z*/x*y + z*x*/y + z*x*y$；

其中，C的逻辑式可以简化为

$C = x*y + x*z + y*z$

下面讨论如何用GAL16V8实现。

（1）将GAL的引脚指派为全加器的输入输出变量。在指派输出变量时一定要考虑哪些GAL16V8引脚具有输出功能。图2-14提供了一种指派方案。图中nc表示没有使用的引脚。

(2) 按规定的格式（参看有关说明）编写如下的逻辑描述文件。

```
PLD16V8
designer: W.B
signature
SAMPLE1
pindef
nc
x y; binary inputs
z; carry input
nc nc nc nc nc gnd
nc nc nc nc nc
S; sum output
C; carry output
nc nc Vc
equation
S =/z*/x*y + /z*x*/y + z*/x*/y + z*x*y*
C =/z*x*y + z*/x*y + z*x*/y + z*x*y*
End
```

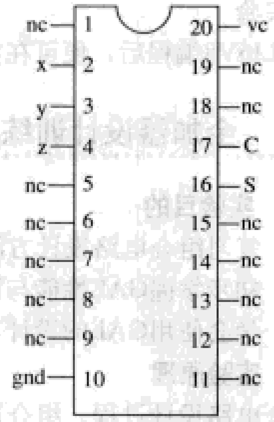


图2-14 GAL16V8引脚指派图

在上面的逻辑式中，*C*的表达式可以用简化式，但从GAL16V8的结构可知，它的每一输出通道允许有七八个与项相加（相或），*C*的逻辑式在简化前后均不超过此结构上的限制，因此不简化也可。

(3) 启动计算机，调用GS程序。使用其中的EDITOR将设计好的逻辑描述文件（这里命名为SAMPLE.PLD）输入计算机，再用其中的COMPILER对它进行编译，生成GAL16V8的熔点图文件SAMPLE.FMP，接着可用其中的SIMULATOR对SAMPLE.FMP进行仿真。如果逻辑功能符合要求可进行下一步。如果编译出现错误或逻辑功能不符要求，便要修改PLD文件，重新进行编译和仿真，直至设计符合要求。

(4) 用编程器将设计好的烧写文件写入GAL16V8。

(5) 在实验箱上接线测试，验证已编程的GAL16V8，实现全加器的功能。

注意：GAL16V8的10脚固定接地线，20脚固定接5V电源，如图2-14所示，不可接错。GAL16V8芯片较贵，使用时要当心。

三、实验预习要求

(1) 仔细阅读2.1节中有关GAL的结构和使用的内容。

(2) 查阅通用编程器的使用方法和标准烧写文件（.JED文件）的产生方法。

(3) 设计一个多数表决逻辑电路，该电路有3个输入端，1个输出端，它的逻辑功能要求是：当有2个以上输入为高电平时，输出为高电平，在其他情况下输出都应是低电平。要求按上例格式编写好PLD文件。

四、实验内容及实验步骤

(1) 在计算机上调用GS软件包，将设计好的PLD文件输入计算机，进行编译、仿真和修改，直至符合设计要求。

(2) 将GS的文本文件转换为FM（Fast-Map软件包）的文本文件，注意如下修改并另存。

①大写第一行（例PLD16V8）。

②第五行开始必须为引脚。

③引脚下一行直接列表达式（删除equation）。

④End前加DESCRIPTION（大写）。

⑤ .OE（大写）=1，改为 .OE=VCC。

⑥电源必须定义为VCC。

（3）将修改后的文本文件装入FM编译软件。选择生成JED文件（标准烧写文件），并存盘。

（4）将编程器与计算机相连，开启编程器电源，打开编程器应用程序。首先选择器件GAL16V8，按编程器上的图形位置和缺口方向将GAL16V8放入对应的插座（注意：位置齐底，缺口向上），按下插座上的锁紧杆。其次通过网络共享，装入自己设计好的JED文件，运行编程（写入）并阅读编程报告。确定烧写无误后，打开插座上的锁紧杆，将GAL芯片取出。

（5）在实验箱上选择20脚的插座，插入已编程的GAL16V8芯片，接上输入和输出，测试全加器的功能表。

（6）注意以下几项。

- 在打开或关闭编程器电源时，GAL芯片一定不要留在编程插座上，否则将损坏芯片。
- GAL16V8后面的（Normal）表示正常的编程参数。只有教师指定的芯片才能使用特殊的编程参数。错误的选择可能导致芯片损坏。
- 以上有关GAL编程操作以及所列注意事项，适用于随后的各次实验训练，一定要通过此次实验搞清楚并掌握。

五、实验报告要求

（1）写出设计思路，画出引脚指派图，列写PLD文件。

（2）总结调试中的心得体会。

实验二 顺序密码锁设计

一、实验目的

（1）复习同步时序逻辑电路的设计方法。

（2）练习将GAL芯片用于实现同步时序逻辑电路。

二、实验原理

1. 同步时序逻辑电路设计简述

时序逻辑电路内部含有存储元件，因此它的输出不仅与当前的外输入有关，而且还与存储元件当前的状态有关。在考虑设计时，可从电路结构上将其中的存储部分框出，余下部分即为组合逻辑电路。如图2-15所示。

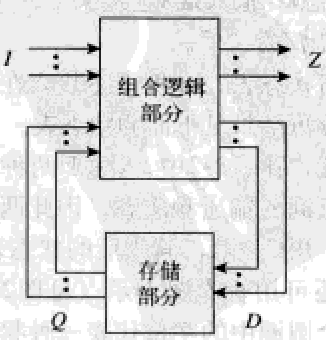


图2-15 时序电路结构

组合电路的输入可分为两部分，一部分为外部输入I，另一部分输入信号来自内部存储元件的输出Q。后者反映了时序电路的内部状态。组合

电路的输出也可分为两部分，一部分接对外输出端Z，另一部分输出信号控制存储部分的输入D。后者决定了时序电路的下一个状态（次态）。

如上所述，时序电路组合部分的逻辑方程可分为如下2组。

（1）输出方程组$Z=Z\ (I,\ Q)$。

（2）驱动方程组$D=D\ (I,\ Q)$。

对于存储部分，因为它们由各类触发器组成，描述它们的状态转换与输入关系的是特征方程组。对于GAL，它的内部只有D触发器，其特征方程组是

$$Q:=D$$

这里，Q是次态输出（即在下一个触发脉冲加入以后的状态输出），D是现态输入。为表示出该方程组两侧时态的差异，在此，Q与D之间的等号用“：=”表示。特征方程组与驱动方程组相结合，可得到状态方程组

$$Q:=D\ (I,\ Q)$$

在状态方程组中，正如上面所述，等号两边的Q也存在时态的差异。

同步时序电路设计大体有如下步骤。

（1）需求分析与描述。

根据功能要求，分析电路需要多少个不同的状态（状态可用文字符号表示）以及各个状态的输入输出情况，设计出相应的电路结构，根据分析结果，画出状态表或状态图。

（2）分析比较状态表中各状态的输入与输出情况，简化状态表。

（3）用触发器的状态组合（可表示成二进制数）取代表中状态的文字符号（状态指派）。

（4）状态指派后，便可根据状态表，列写输出方程组和触发器的状态方程组。

（5）根据设计好的逻辑方程组，编写GAL的逻辑描述文件（.PLD文件），用GAL加以实现。

实现过程参看实验一中的有关说明。

例：设计一个“111”序列检出器，它的功能是，当输入X连续3次或以上为1时，输出Z为1，否则Z为0。

（1）根据题意，可设想此一序列检出器有如下几种状态。

- 状态A：初始状态，此时输入X的信号电平都是“0”，输出Z也是“0”。
- 状态B：输入端X只来了1个“1”，输出Z为“0”。
- 状态C：输入端X连续来了2个“1”，输出Z为“0”。
- 状态D：输入端X连续来了3个或更多的“1”，输出Z为“1”。

列写状态表，分析上述状态在不同输入时的次态，分析结果见下面的状态表。

注意，在表2-2中，C和D两种状态在输入相同时有相同的输出和次态，因此两种状态可简化为一种，用C表示，简化后的状态表见表2-3。此状态表还可用状态图表示（见图2-16）。状态图中，每个圆圈中的字母代表一种状态，连线旁边的数字是“输入/输出”，表示状态变迁的条件和

表2-2 序列检出器状态表

输入X	现态	次态	输出Z
0	A	A	0
1	A	B	0
0	B	A	0
1	B	C	0
0	C	A	0
1	C	D	1
0	D	A	0
1	D	D	1

结果。

表2-3 简化后的状态表

输入X	现态	次态	输出Z
0	A	A	0
1	A	B	0
0	B	A	0
1	B	C	0
0	C	A	0
1	C	C	1

（2）状态指派。简化后只有3个状态，因此用两个D触发器Q1和Q2即可。A、B和C状态可分别指派为00、01和10，如表2-4所示。

表2-4 编码后的状态表

输入及现态			次态和输出		
X	Q1	Q2	Q1	Q2	Z
0	0	0	0	0	0
1	0	0	0	1	0
0	0	1	0	0	0
1	0	1	1	0	0
0	1	0	0	0	0
1	1	0	1	0	1

根据状态表可列出如下状态方程和输出方程

$$Q1: = X*/Q1*Q2 + X*Q1*/Q2;$$

$$Q2: = X*/Q1*/Q2;$$

$$Z = X*Q1*/Q2;$$

有了上面的方程后，就可用实验一中所列步骤与方法，利用GAL实现此电路。

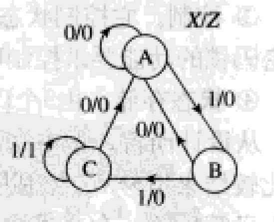


图2-16 简化后的状态图

2. 顺序密码锁

（1）顺序密码锁工作原理

图2-17为顺序密码锁外部引线图，其中I1~I6为预设的密码输入，KEY为开锁码输入端，R为复位信号输入，顺序密码锁对外输出信号有二个，一是开锁信号（UL），一是报警信号（WARN）。

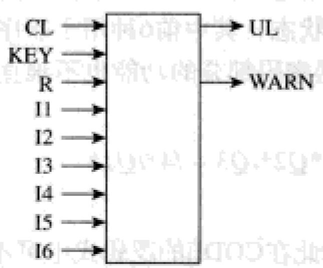


图2-17 顺序密码锁外部引线图

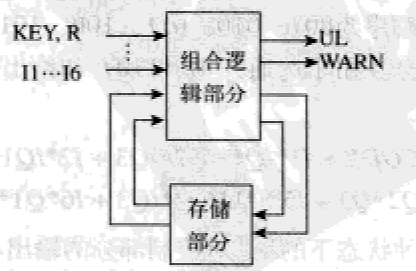


图2-18 顺序密码锁的逻辑结构图

顺序密码锁的工作原理为：在CL脉冲作用下，KEY端顺序输入高低电平序列（开锁码序列）。如果该序列与预设定的电平序列（其顺序与I1~I6等6个输入端上预先设定的电平相同）完全一致，则UL端出现高电平，LED指示灯亮，表示锁已打开。若输入电平序列与预设的序列不完全一致，则WARN端出现高电平，报警指示灯亮，需要复位才能再次输入。

（2）顺序密码锁的设计

从上述工作原理可知，顺序密码锁需顺序识别6位密码，因此，至少要用3个触发器记忆这6种状态。如果简单地将它分解为组合逻辑与存贮元件两部分，如图2-18所示，那么该组合电路将有8个外部输入和3个内部状态信号，输入线总计有11根，因此输入变量多，设计难度较大。所以，在进行逻辑设计以前，一般都应尽可能地将其分解成功能模块以简化设计。图2-19就是按此思路得出的顺序密码锁内部框图。

图2-19内部有4个框，它们分别是密码、比较、控制与状态存储，除状态存储由D触发器构成外，其他都是组合逻辑电路。各框功能分别讨论如下。

① 密码。根据状态的顺序变化（Q1、Q2和Q3为状态信号），它的输出端CODE将顺序输出不同的密码电平（I1~I6）。

② 比较。在时钟脉冲控制下，开锁码与密码串逐位进行比较，相同则输出（X）高电平，否则输出低电平。

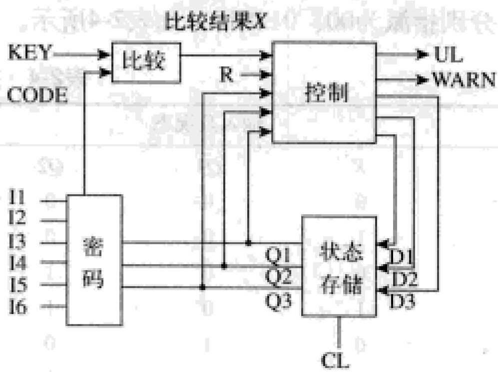


图2-19　顺序密码锁内部框图

③ 控制。它控制状态的转换，并根据顺序密码锁的功能要求控制UL和WARN信号的输出。

④ 状态存储。由3个D触发器构成，在时钟和D1、D2和D3的控制下改变状态。

从设计而言，图中密码和比较两部分的作用是将I1~I6和KEY等7个外输入信号简化为一个比较结果信号X。控制框和状态存贮框构成一个典型的时序电路结构。对控制框而言，外输入已简化为一个变量X，因此不难用上面介绍的方法设计。下面就如何在GAL内部实现密码和比较两个部分进行说明。这两部分都是组合逻辑电路，它们都只有一根输出线，因此要分别占用一个输出通道。比较部分的逻辑式较简单，可根据功能要求直接写出

$$X = KEY*CODE + /KEY*/CODE$$

密码部分虽输入线较多，但可根据其功能分成两组，I1~I6是设置的密码，在不同的状态下CODE电平只与其中一根线的电平相同，状态由Q1、Q2和Q3决定，设初始状态指派为000，以下顺序为001、010、011、100、101、110和111共8种状态，其中前6种用于顺序选码，后两种分别用于选通开锁（110）和给出报警（111）。根据密码部分的功能也不难直接写出逻辑式。

$$CODE = I1*/Q1*/Q2*/Q3 + I2*/Q1*/Q2*Q3 + I3*/Q1*Q2*/Q3 + I4*/Q1*Q2*Q3 + I5*Q1*/Q2*/Q3 + I6*Q1*/Q2*Q3$$

后两种状态下的X值对控制部分的输出不应产生影响，因此在CODE的逻辑式中可不予考虑。

由于此芯片内要用触发器，因此它的组合逻辑通道构型为“F”，所以要对它们的三态控

制项（OE项）单列方程

$X.OE=1$；

$CODE.\ OE=1$

控制部分和存贮部分的设计可按时序电路设计方法进行。图2-20是顺序密码锁的参考状态图，图中所用状态指派值同上，具体设计由读者自己完成。

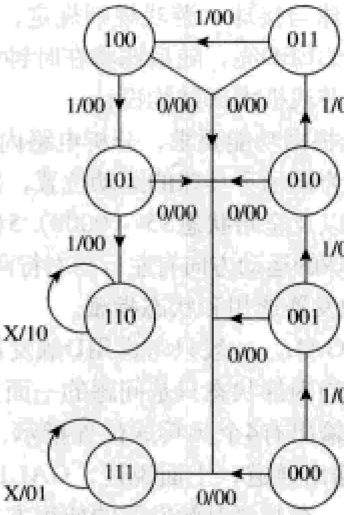


图2-20 顺序密码锁的参考状态图

三、实验预习要求

（1）画出6位顺序密码锁的状态表，并设计出它的逻辑式。

（2）写出用一片GAL16V8实现顺序密码锁的PLD文件。

四、实验内容及实验步骤

（1）在计算机上输入自己编写的PLD文件，进行编译、仿真和修改，直到符合设计要求。由于电路内有组合逻辑通道，在仿真时如果遇到问题，要注意灵活应用仿真操作键S、I和A，以确定问题所在。

（2）用编程器对GAL16V8编程。

（3）在实验箱上进行实验。实验箱上设有单脉冲和高低电平开关，单脉冲可用作时钟，高低电平开关可用于输入开锁码，密码电平可用连线或电平开关设置。

注意事项与上一个实验相同。

五、实验报告要求

（1）写出设计的PLD文件和调试过程。

（2）GAL芯片提供了灵活设计数字电路的可能性。如果读者对实验有所改进，请着重讨论改进部分。

实验三 乒乓游戏机设计

一、实验目的

（1）进一步掌握同步时序电路的设计方法。

（2）学习怎样根据GAL芯片特点，选用不同的电路结构。

二、实验原理

1. 乒乓游戏机设计

（1）功能要求

图2-21为乒乓游戏机框图，图中以4个发光二极管（LED）模拟乒乓球的运动，每次点亮一个LED来表示球运行的当前位置，在时钟脉冲CL作用下它将自动运行到下一个位置。以两个电平开关A和B作为球拍，两个游戏者各控其一，按下开关表示击球，只有当球到达最靠边的位置时击球才算接中。R

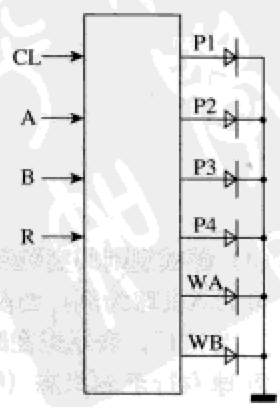


图2-21 乒乓游戏机框图

出现高电平“1”后，乒乓球位置状态应是S5（0000），LED全暗，以此作为系统初始状态。为区分发球与接球，游戏规则规定，在发球时，除A或B应击球外，R也应为1。发球时，发球方一侧的LED亮，随后球将在时钟作用下向对方运动。

（2）游戏机内部结构设计

① 先根据功能要求，分析电路内部需要有多少不同的状态。

- 为显示乒乓球的运动位置，需要有S1（1000）、S2（0100）、S3（0010）、S4（0001）以及全暗状态S5（0000）5种不同的状态。
- 球的运动方向有左行与右行两种，因而又增添两种状态。所以至少需要7种不同状态。

② 触发器选用和状态指派。

对于GAL，一般只考虑用D触发器，为实现7种不同的状态，用3个触发器就可以了。但实现所需的内部状态只是问题的一面，该电路还要实现对外输出。

这些输出有4个乒乓球位置指示、2个胜负指示，以及表示状态用的3个触发器，即一共要有9个输出通道，上面说过，GAL16V8只有8个输出通道，为了能用一片GAL16V8实现乒乓游戏机，有必要对触发器的使用方式加以变通。

下面采用的方案是，用4个触发器输出通道直接控制乒乓球的位置指示，同时也可用它们记忆有关状态，由于乒乓球位置指示只有5种，因此这4个触发器P1、P2、P3和P4只能用来实现5种状态，左右行控制需另设一触发器DI实现。根据此方案，5个触发器加上2个胜负指示，一共只需7个输出通道。也就是说，可用一片GAL16V8实现游戏机电路。它的内部结构如图2-22所示。

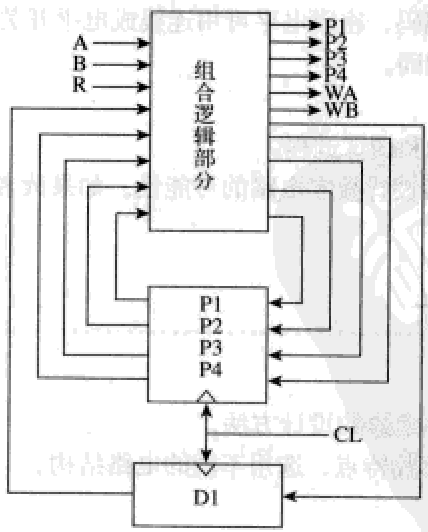


图2-22 游戏机电路的内部结构图

（3）游戏规则的实现和状态表。

① 以A发球为例。当A和R都为高电平时，P1~P4的状态应为S1（1000），此时D1的状态为“1”，表示球在时钟脉冲作用下向右运动。

② 球右行至S4状态（0001）时，如果B没有及时击球，在时钟作用下将进入S5（0000）状态，此时WA亮WB暗，表示A胜B负。

③ 如果B及时击球，球的位置将进入S3状态（0010），DI的状态也转换为“0”，在时钟作用下球向左运动。

④ 球左行及以后可能出现的状态转换请同学自行补充。

根据以上规则，可画出如表2-5所示部分状态表，仅供参考，请补充。

表2-5 乒乓游戏机部分状态表

输入			现态					次态					输出	
R	A	B	DI	P1	P2	P3	P4	DI	P1	P2	P3	P4	WA	WB
1	0	0	×	×	×	×	×	×	0	0	0	0	0	0
1	1	0	×	×	×	×	×	1	1	0	0	0	0	0
1	0	1	×	×	×	×	×	0	0	0	0	1	0	0
1	1	1	×	×	×	×	×	×	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	0	1	0	0	0	0
0	0	0	1	0	1	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	1	0	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	1	0	0	0	0	1	0
0	×	×	1	0	0	0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

三、实验预习要求

（1）将以上游戏规则及状态表补充完整。

（2）根据状态表设计游戏机的逻辑方程组，指派GAL16V8的引脚，写出PLD文件。

（3）在设计组合逻辑输出表达式时，可以选用现态变量也可以选用次态变量作为输入，如何选择应根据设计要求。

四、实验内容及实验步骤

（1）在计算机上输入自己编写的PLD文件，进行编译、仿真和修改，直到符合设计要求。

由于电路内有组合逻辑通道，在模拟时如果遇到问题，要注意灵活应用多步仿真提供的多种仿真方式以确定问题所在。

（2）用编程器对GAL16V8编程。

（3）在实验箱上进行实验，实验箱上设有连续脉冲和高低电平开关，连续脉冲可用作时钟，高低电平开关或单脉冲可用于给出R和A、B输入信号。

注意事项与上一个实验相同。

五、实验报告要求

（1）写出设计的PLD文件和调试过程。

（2）GAL芯片提供了灵活设计数字电路的可能性。如果读者对实验有所改进，请着重讨论改进部分。

实验四 交通灯控制电路设计

一、实验目的

（1）学习在同步时序电路中处理异步输入。

（2）较独立地完成本实验电路的设计。

二、实验原理及说明

交通灯控制电路设计说明

（1）功能要求。

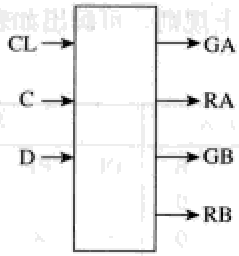


图2-23 交通灯控制框图

图2-23为交通灯控制框图，图中RA和GA为十字路口东西向（方向A）红绿灯，RB和GB为南北向（方向B）红绿灯。CL为内部时钟脉冲，它的频率为1次/min，在它的作用下，交通灯交替工作，使A向和B向轮流放行1min。C和D为设置在路边的按钮，当行人欲横过街道时，可按下C或D按钮，控制器接收到此信号，就会在当前的1 min周期结束后，让全部的红色交通灯亮，也持续1 min，让行人通过，之后再回到原工作循环。

（2）内部结构设计提示。

① 电路内部状态分析。

- 状态1：A向通行。
- 状态2：B向通行。
- 状态3：A向通行后转行人通过，以后转B向通行。
- 状态4：B向通行后转行人通过，以后转A向通行。

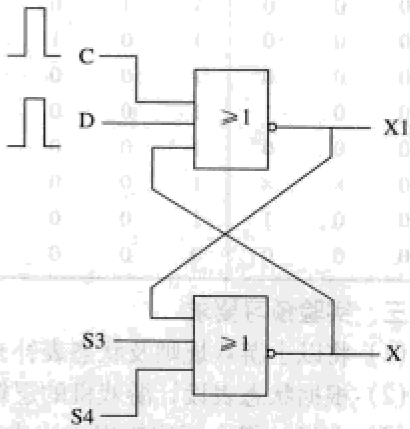


图2-24 RS触发器电路

共可列出4种状态。

② 外部输入信号的处理。该控制器的输入是“行人通过请求”信号，它由C或D按钮产生，由于时钟频率太低，行人按下C或D的持续时间一般小于1 min，且不可能要求它与时钟同步（异步信号），所以内部需另设电路记忆由C或D来的信号。在GAL内部，D触发器的状态转换均与时钟同步，因而不可能用它们记忆外输入的请求信号。这就需要用图2-24所示的RS触发器电路。RS触发器的输出信号为X，X又作为控制部分的输入。为实现预定交通控制功能，该RS触发器的置位信号应是C或D，复位信号应来自状态3（S3）或4（S4）。由图可见，此RS触发器由2个组合逻辑输出通道（F型）X和X1构成。

综上所述，交通控制器的内部框图如图2-25所示。

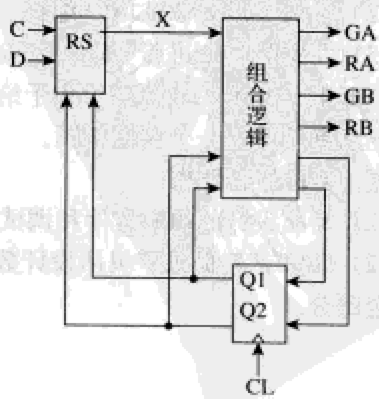


图2-25 交通控制器的内部框图

三、实验预习要求

(1) 设计交通控制电路的状态图与状态表。

(2) 根据状态表设计逻辑方程组，指派GAL16V8的引脚，写出PLD文件。

注意：*对于F型通道，在逻辑方程组中要列写.OE控制方程。*

四、实验内容及实验步骤

(1) 在计算机上输入自己编写的PLD文件，进行编译、仿真和修改，直到符合设计要求。

由于电路内有组合逻辑通道，在仿真时如果遇到问题，要注意灵活应用多步仿真提供的多种仿真方式以确定问题所在。

(2) 用编程器对GAL16V8编程。

(3) 在实验箱上进行实验，实验箱上用单脉冲模拟时钟，高低电平开关或其他单脉冲源可用作信号C和D。

注意事项与上一个实验相同。

五、实验报告要求

(1) 写出设计的PLD文件和调试过程。

(2) 总结处理异步输入的体会。

(3) GAL芯片提供了灵活设计数字电路的可能性。若读者对实验有所改进，请着重讨论改进部分。

2.2 在系统编程设计训练

2.2.1 ispEXPERT简介

运用大规模、超大规模可编程逻辑器件与电子设计自动化（EDA）技术是现代数字电子技术发展的潮流，将其引入高校、研究所和电子产品生产企业势在必行。美国Lattice公司推出的ISP技术为电子系统设计提供了功能完善的设计手段，特别适合教学、科研和产品开发。

ISP技术使用户能够在不从电路板上拆下芯片的情况下，改变芯片的逻辑内容。这种技术能大大缩短电子系统设计周期，简化生产流程，降低生产成本，并可在现场对系统进行逻辑重构和升级。ISP技术的发明，使硬件随时能够改变组态，实现了硬件设计软件化，革命性地改变了电子系统设计的传统概念和方法。

Lattice公司推出的ispEXPERT数字系统设计软件，是一套完整的EDA软件。设计输入可采用原理图、硬件描述语言和混合输入三种方式。能对所设计的数字电子系统进行功能仿真和时序仿真。ispEXPERT 编译器是此软件的核心，能进行逻辑优化、将逻辑映射到器件，以及自动完成布局与布线并生成编程所需的熔丝图文件。软件支持所有Lattice公司的ispLSI器件。

软件主要特征如下。

(1) 输入方式

• 原理图输入

• ABEL-HDL输入

- VHDL输入
- Verilog-HDL输入

(2) 逻辑仿真

- 功能仿真
- 时序仿真
- 静态时序分析

(3) 编译器

- 结构综合、映射、自动布局和布线

(4) 支持的器件

- 含有宏库，有500个宏元件可供调用
- 支持所有ISP器件

(5) 下载软件

- ISP菊花链下载软件

2.2.2 在系统编程的操作方法

这里介绍ABEL语言和原理图混合输入的操作方法。

建立一个简单的ABEL- HDL语言输入的设计，并且将其与原理图进行合并，以层次结构的方式，画在顶层的原理图上。然后对这个完整的设计进行仿真和编译，最后适配到ispLSI器件中。

1. *启动*ispEXPERT System

单击Start=>Programs=>Lattice Semiconductor=>ispEXPERT System菜单，屏幕上的项目管理器应该如图2-26所示。

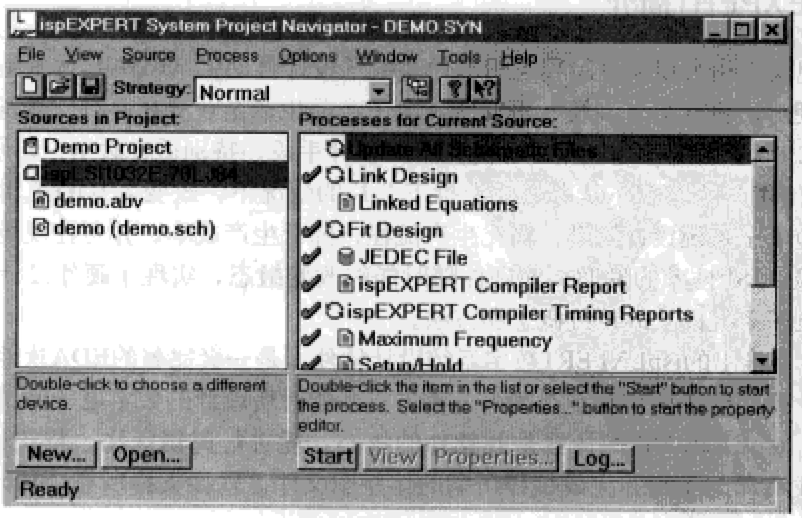


图2-26 启动ispEXPERT System

2. *建立顶层的原理图*

① 选择1032E器件，从菜单条上选Source。

② 选择New...

③ 在对话框中选Schematic，并按OK。

④ 选择路径c:\ examples，然后在文本框中输入文件名top.sch，并按OK。

⑤ 现在就进入了原理图编辑器。

⑥ 调用已创建的元件符号。选择Add菜单中的Symbol项，这时会出现Symbol Libraries对话框，选择Local的库，注意到在下部的文本框中有一个叫demo的元件符号，这就是已自行建立的元件符号。

⑦ 选择demo元件符号，放到原理图上的合适位置。

3. 建立内含ABEL语言的逻辑元件符号

现在要为ABEL-HDL设计文件建立一个元件符号。只要知道了接口信息，就可以为下一层的设计模块创建一个元件符号。而实际的ABEL设计文件可以在以后再完成。

① 在原理图编辑器里，选择ADD菜单里的New Block Symbol... 命令。

② 出现一个对话框，提示输入ABEL模块名称及其输入信号名和输出信号名。请按照图2-27所示输入信息。

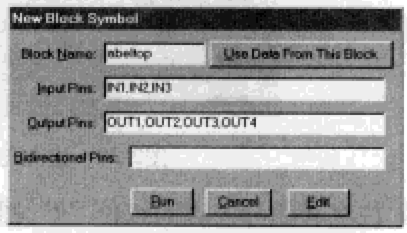


图2-27 ABEL模块信息

③ 完成信号名的输入，按Run按钮，就会产生一个元件符号，并将其放在本地元件库中。同时元件符号还黏在光标上，随之移动。

④ 把这个符号放在demo符号的左边。

⑤ 单击鼠标右键，就会显示Symbol Libraries的对话框。注意，abeltop符号出现在Local库中。

⑥ 关闭对话框。原理图应该如图2-28所示。

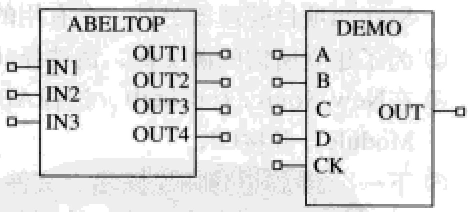


图2-28 含ABEL语言的逻辑元件符号原理图

4. 完成原理图

现在添加必需的连线、连线名称以及I/O标记，来完成顶层原理图，如图2-29所示。画完后，存盘再退出。

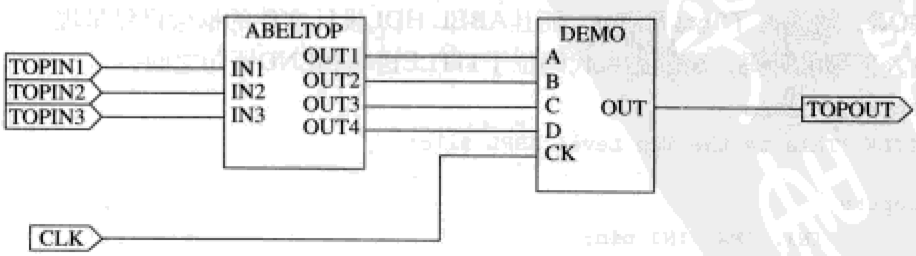


图2-29 顶层原理图

5. 建立ABEL-HDL源文件

现在需要建立一个ABEL源文件，并把它链接到顶层原理图对应的符号上。项目管理器

简化了这些步骤。

① 当前的管理器应该如图2-30所示。

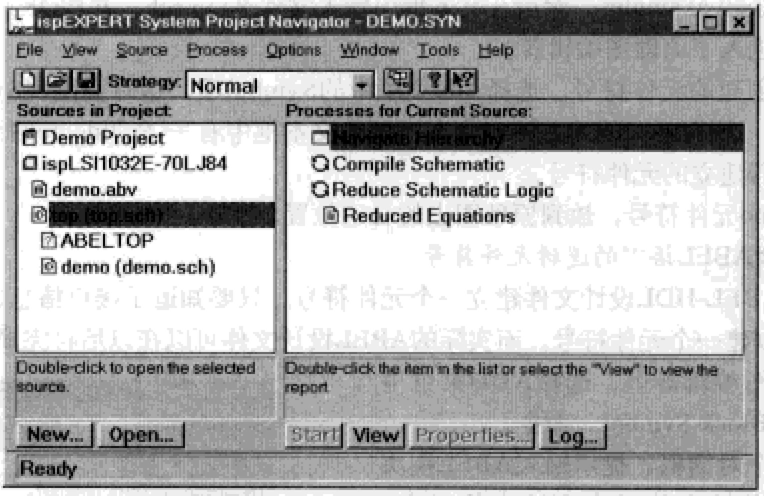


图2-30 建立与原理图对应符号相链接的源文件

② 注意abeltop左边的红色“?”图标。这意味着目前这个源文件还是个未知数，因为还没有建立它。同时也要注意源文件框中的层次结构，abeltop和demo源文件位于top原理图的下面并且偏右，这说明它们是top原理图的底层源文件。这也是ispEXPERT System项目管理器另外一个有用的特点。

③ 为了建立所需的源文件，请选择abeltop，然后选择Source菜单中的New... 命令。

④ 在New Source对话框中，选择ABEL-HDL Module，并按OK。

⑤ 下一个对话框询问模块名、文件名和模块的标题。为了将源文件与符号相链接，模块名必须与符号名一致，而文件名没有必要与符号名一致。但为了简单，可以给它们取相同的名字。按图2-31所示，填写相应的栏目。

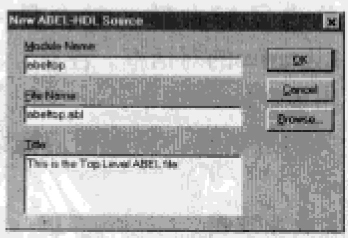


图2-31 输入新的模块信息

⑥ 按OK，就进入了Text Editor，而且ABEL HDL设计文件的框架已经呈现。

⑦ 输入下列的代码。确保输入代码位于TITLE语句和END语句之间。

```
MODULE abeltop
TITLE 'This is the Top Level ABEL file'

"Inputs
        IN1, IN2, IN3 pin;
"Outputs
        OUT1, OUT2, OUT3, OUT4      pin;

Equations
        OUT1=IN1 & ! IN3;
```

```
        OUT2=IN1 & ! IN2;
        OUT3=!IN1 & IN2 & IN3;
        OUT4=IN2 & IN3;
END
```

⑧ 完成后，选择File菜单中的Save命令。

⑨ 退出文本编辑器。

⑩ 注意项目管理器中abeltop 源文件左边的图标已经改变了。这意味着已经有了一个与此源文件相关的ABEL文件，并且已经建立了正确的链接。

6. 编译ABEL-HDL

① 选择abeltop源文件。

② 在处理过程列表中，双击Reduce Logic过程。会看到项目管理器在执行 Reduce Logic过程之前，先去执行Compile Logic过程。当处理过程结束后，项目管理器应该如图2-32所示。

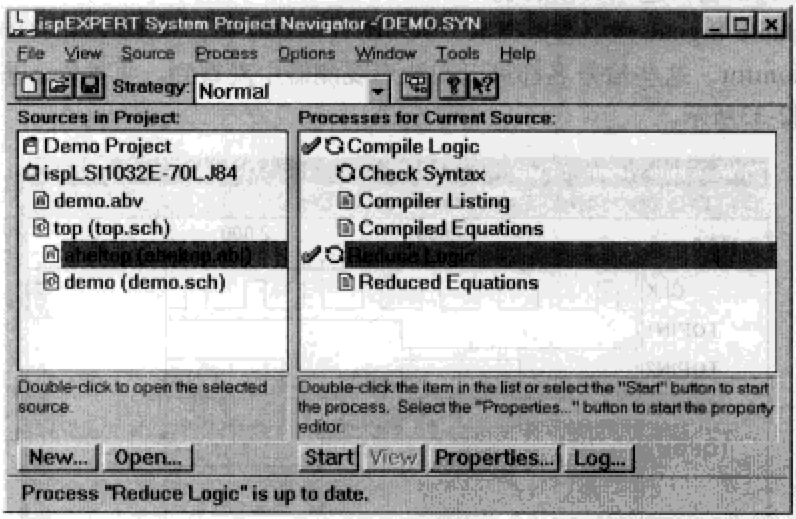


图2-32 编译ABEL-HDL

7. 仿真

现在可以对整个设计进行仿真。为此，需要一个新的测试矢量文件。在这个例子中只需要修改当前的测试矢量文件。

① 双击demo.abv源文件，就会出现文本编辑器。

② 按照以下内容修改测试矢量文件。

```
module demo;

c, x = .c., .x.;

CLK, TOPIN1, TOPIN2,TOPIN3, TOPOUT PIN;

TEST_VECTORS

([CLK, TOPIN1, TOPIN2, TOPIN3]->[TOPOUT])
  [ c  ,  0  , 0  ,  0 ]->[ x ];
```

```
[ c , 0 , 0 , 1 ]->[ x ];
[ c , 0 , 1 , 0 ]->[ x ];
[ c , 0 , 1 , 1 ]->[ x ];
[ c , 1 , 0 , 0 ]->[ x ];
[ c , 1 , 0 , 1 ]->[ x ];
[ c , 1 , 1 , 0 ]->[ x ];
[ c , 1 , 1 , 1 ]->[ x ];
END
```

③ 完成后，存盘退出。

④ 选择测试矢量源文件，双击Functional Simulation过程，进行功能仿真。

⑤ 进入Simulation Control Panel窗口。按Windows=> Waveform Viewer窗口，打开波形观测器，准备查看仿真结果。

⑥ 为了看波形，必须在Simulation Control Panel窗口中按Debug钮，使Simulation Control Pane窗口进Debug模式。

⑦ 在Available Signals栏中选择CLK、TOPIN1、TOPIN2、TOPIN3和TOPOUT信号，并且按Monitor。这些信号名都可以在波形观测器中观察到。再按Run进行仿真，其结果如图2-33所示。

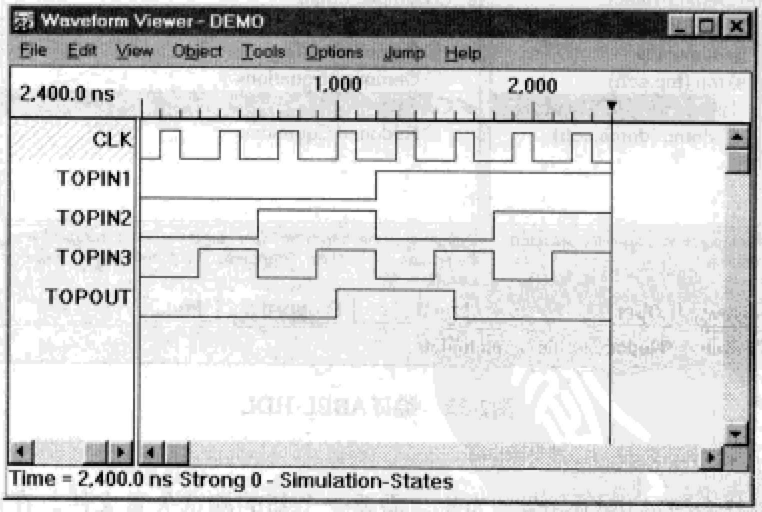


图2-33 仿真示意图

⑧ 在步骤④中，如双击Timing Simulation过程，即可进入时序仿真流程，以下仿真步骤与功能仿真相同。

8. 把设计适配到Lattice器件中

现在已经完成了原理图和ABEL语言的混合设计及其仿真。接下来将设计放入Lattice ispLSI器件中。选择好器件后，可以直接执行下面的步骤。

① 在源文件窗口中选择ispLSI1032E-70LJ84器件作为编译对象，并注意观察对应的处理过程。

② 双击处理过程Fit Design。这将使项目管理器完成对源文件的编译，然后连接所有的源文件，最后进行逻辑分割、布局和布线，将设计适配到所选择的Lattice器件中。

③ 当这些都完成后，可以双击ispEXPERT Compiler Report，查看一下设计报告和有关统计数据。

④ 现在已经完成了设计例子，并且掌握了ispEXPERT System的主要功能。

9. 层次化操作方法

层次化操作是ispEXPERT系统项目管理器的重要功能，它能够简化层次化设计的操作。

① 在项目管理器的源文件窗口中，选择最顶层原理图“top.sch”。此时在项目管理器右边的操作流程清单中必定有Navigation Hierarchy过程。

② 双击Navigation Hierarchy过程，即会弹出最顶层原理图“top.sch”。

③ 选择View菜单中的Push/Pop命令，光标就变成十字形状。

④ 用十字光标单击顶层原理图中的abeltop符号，即可弹出描述abeltop逻辑的文本文件abeltop.abl。此时可以浏览或编辑ABEL HDL设计文件。浏览完毕后用File菜单中的Exit命令退回顶层原理图。

⑤ 用十字光标单击顶层原理图中的demo 符号，即可弹出描述demo逻辑的底层原理图demo.sch。此时可以浏览或编辑底层原理图。

⑥ 若欲编辑底层原理图，可以利用Edit菜单中的Schematic命令进入原理图编辑器。编译完毕后用File菜单中的Save和Exit命令退出原理图编辑器。

⑦ 底层原理图浏览完毕后用十字光标单击图中任意空白处，即可退回上一层原理图。

⑧ 若某一设计为多层次化结构，则可在最高层逐层进入其底层，直至最底一层，退出时亦可以从最底层逐层退出，直至最高一层。

⑨ 层次化操作结束后用File菜单中的Exit命令退回项目管理器。

10. 锁定引脚的另一种方法

引脚的锁定除了在原理图中定义I/O Pad的属性外，还可用引脚锁定文件（*.ppn）的形式来实现。其操作方法如下。

① 按照规定的格式建立引脚锁定文件，如下所示。

```
【引脚名称】          【引脚属性】          【引脚编号】

以下为引脚锁定文件实例 ------ PIN_LOCK.PPN:
TOPIN1                IN                    26
TOPIN2                IN                    27
TOPIN3                IN                    28
TOPOUT                OUT                   29
CLK                   IN                    30
```

② 在源文件窗口中选择一种具体的器件，如ispLSI 1032E-70LJ84，然后在处理过程窗口中选择Fit Design 功能，此时位于窗口下方的Properties按钮被激活。

③ 按下Properties，打开控制参数编辑对话框。在控制参数编辑对话框中找到Pin File Name 行，单击该行使之进入编辑方式，然后在输入栏中键入参数文件名称PIN_LOCK.PPN，并按下“√”确认之。

④ 按下控制参数对话框中的Close，关闭对话框。

至此就可以用引脚锁定文件来控制适配器的编译，而原理图中原来锁定的引脚无效。锁定效果在器件编译完后通过ispEXPERT Compiler Report反映。

2.2.3 ABEL-HDL语言介绍

1. ABEL-HDL模块的基本结构

① 头部

(a) MODULE语句

关键字：MODULE

语法：module 模块名

模块内容

end 模块名

例：module my_example

……

C=B + A;

……

end my_example

(b) FLAG语句

关键字：FLAG

语法：flag '参数1'，'参数2'，…，'参数n'；

（参数为ABLE软件的处理参数，有化简级别、跟踪级别、校验和参数、未用熔丝参数等形式）

flag '-r2'，'-t3'

（r-n软件对源文件的化简级别，t-n为跟踪级别）

(c) INTERFACE语句

关键字：INTERFACE

在层次说明中使用

(d) TITLE语句

关键字：TITLE

语法：title ['字符串']

例：title 'memory decode Jean Designer Data I/O Corp 24 Feb 2004'

② 说明部

(a) DECLARATIONS语句

关键字：DECLARATIONS

语法：DECLARATIONS

若说明部紧跟头部之后，则该语句可省略。

(b) 器件说明

关键字：DEVICE

语法：器件标志符 DEVICE 实际器件;

例：U1 device 'MACH211'；

(c) 常量说明

关键字："="

语法：标志符[，标志符，…] = 表达式[，表达式，…]；

例：X=.X..；

Addr=[1，0，1]；

(d) 信号说明

(i) 引脚说明（模块对外的信号）

关键字：PIN

语法：引脚名[，引脚名，…] PIN [引脚号[，引脚号，…]] [ISTYPE '属性']；

例：a，b，c pin 3，8，10；

(ii) 节点说明（模块内部的信号）

关键字：NODE

语法：标志符[，标志符，…] = 表达式[，表达式，…]；

例：x，y，z node；

(iii) 属性说明（对输出信号的说明，表2-6）

关键字: ISTYPE

语法: 信号 [，信号，…] ISTYPE '属性'；

例：o1，o2，o3 pin istype 'reg_d，buffer'；

表2-6 属性说明

属性	说明
Invert	目标器件中寄存器输出到引脚输出之间有反向器
Pos	正极性
collapse	优化时此信号可能去掉
Dc	未指定的逻辑为任意态
Xor	目标器件中的XOR门
Com	组合信号
feed-pin	由引脚反馈
feed-or	由或门反馈
reg-g	G型（时钟使能）寄存器
reg-rs	RS型寄存器
buffer	目标器件中寄存器输出到引脚输出之间无反向器
neg	负极性
keep	优化时此信号不能去掉
retain	不对输出进行最小化优化保留冗余乘积项
latch	锁存输入脚
reg	寄存器信号
feed-reg	由寄存器反馈
reg-d	D寄存器
reg-jk	JK型寄存器
reg-t	T寄存器

(e) 层次说明

(i) 低层模块接口说明

关键字：INTERFACE

语法：MODULE　模块名

INTERFACE（输入/集合[=端口值]->输出/集合:>双向/集合）；

例：module and1

interface　(ia1，ia2，ia3->oa1）；

Title　'3 inputs and gate'；

(ii) 顶层模块接口说明

关键字：INTERFACE

语法：低层模块名INTERFACE（输入[=值]->输出/集合:> 双向/集合)；

(iii) 功能块语句

关键字：FUNCTIONAL_BLOCK

语法：例化名 FUNCTIONAL_BLOCK 模块名；

例：and1 interface　(ia1，ia2，ia3->oa1）；

and1_1 fuctional_block and1;

(f) 符号状态说明

(i) 状态寄存器说明

关键字：STATE_REGISTER

语法：状态寄存器标志 STATE_REGISTER [ISTYPE　'属性'];

(ii) 状态说明

关键字：STATE

语法：状态标志符[，状态标志符，…] STATE [状态值 [，状态值，…]];

(g) 宏说明

关键字：MACRO

语法：宏标志符　MACRO [哑变量，[哑变量]，…] { 块 };

在源文件中使用宏可避免重复写一些功能块的代码。

(h) 库说明

关键字：LIBRARY

语法：LIBRARY　　'库名'

将指定的ABLE-HDL库文件插入到当前文件中。

③ 逻辑描述部

(a) 逻辑等式法

关键字：EQUATIONS

语法：EQUATIONS　[in 器件名] 信号名[?]=表达式; 信号名[?]:=表达式；

说明：该语法中状态转换用WHEN-THEN-ELSE 语句。

例：EQUATIONS

q := a & load # !q.fb & ! load;

(b) 真值表法

关键字：TRUTH_TABLE

语法：TRUTH_TABLE（输入->输出）

TRUTH_TABLE（输入:>寄存器输出[->输出])

例：TRUTH_TABLE （[a， q.fb， load] :> q）

[0， 0， 0] :> 1;

[0， 1， 0] :> 0;

[1， 0， 0] :> 1;

[1， 1， 0] :> 0;

[0， 0， 1] :> 0;

[1， 0， 1] :> 1;

[0， 1， 1] :> 0;

[1， 1， 1] :> 1;

（c）状态描述法

关键字：STATE_DIAGRAM

语法：STATE_DIAGRAM 状态寄存器 [-> 状态输出]

[STATE 状态表达式 : [逻辑等式] [逻辑等式]··· 状态转换···]

说明：该法中状态转换用IF-THEN-ELSE、CASE或GOTO语句实现，在这些语句后可选用WITH语句。

附：点扩展名

语法：信号名.点扩展名

说明：在逻辑描述中使用点扩展名能更精确地描述一个电路的行为。与器件无关的点扩展名见表2-7。

表2-7 与器件结构无关的点扩展名

点扩展名	说明	点扩展名	说明
.ACLR	异步复位	.J	JK触发器的J端
.ASET	异步置位	.K	JK触发器的K端
.CLK	边沿触发器的时钟	.LD	锁存器D端
.CLR	同步复位	.LE	锁存器锁存允许端
.COM	不经寄存器的组合反馈	.LH	锁存器锁存允许H端
.FB	从寄存器输出反馈	.PR	寄存器预置位
.OE	输出允许	.Q	寄存器反馈
.PIN	从输出引脚反馈	.R	RS寄存器R端
.SET	同步置位	.RE	寄存器复位
.AP	异步寄存器预置位	.S	RS寄存器S端
.AR	异步寄存器复位	.SP	同步寄存器预置位
.CE	门控时钟触发器的时钟允许	.SR	同步寄存器复位
.D	D型触发器D端	.T	T型触发器T端
.FC	触发器模式控制		

例：Q1.CLK = PULSE;

Q2.D = A1 & A2 # A3;

④ 测试向量部

（a）测试向量

关键字：TEST_VECTORS

语法：TEST_VECTORS [注释] (输入 -> 输出)

例：test_vectors ([A, B] -> [C, D])

[0, 0]->[1, 1];

[0, 1]->[1, 0];

[1, 0]->[0, 1];

[1, 1]->[0, 0];

(b) 跟踪语句

关键字：TRACE

语法：TRACE (输入 -> 输出)；

说明：用于控制模拟时显示哪些输入和输出，可放在测试向量之前或测试向量中。

⑤ 结束部

关键字：END

语法：END 模块名

2. ABEL_HDL语言的基本语法

① 标识符

标识符是一个名字，它同其他语言中的标识符含义相同。在ABEL-HDL中，它标识器件名、功能块、器件引脚或节点、集合、输入/输出信号等，它区分大小写，且不能与关键字重名。

② 关键字

关键字是特殊的标识符，其功能说明见表2-8。它留作ABEL-HDL语言专用。它们指示出ABEL-HDL语言做哪些事情。关键字不分大小写。

表2-8 关键字功能说明

关键字	功能说明	关键字	功能说明
Module	模块开始语句	End	模块结束语句
Node	节点说明语句	Pin	引脚说明语句
Title	标题语句	Device	器件说明语句
Equations	逻辑方程	State	状态说明语句
Property	特征说明语句	Declarations	声明段关键字
Test_vectors	测试向量	Truth_table	真值表
Functional_block	功能模块说明语句	Fuses	熔丝状态说明语句
Async_reset	异步复位状态描述语言	Sync_reset	同步复位状态描述语句
Interface	功能模块接口语句	State_diagram	状态图关键字
Goto	无条件转移语句	Case-Endcase	条件选择语句
If-Then-Else	条件转移语句，只能用于状态图输入	When-Then-Else	条件转移语句，只能用于方程中
With	转移方程语句	Wait	等待关键字
State_register	状态寄存器说明语句	Options	控制选项定义语句
Istype	属性说明语句	Cycle	周期关键字
Xor_factors	异或因子关键字	Trace	跟踪选项关键字
Library	库引用语句	Macro	宏说明语句

标识符最长31个字符，第一个字符必须是字母或下划线，其他字符可以是大写字母、小写字母、数字、波浪线和下划线。标识符中不能包含句点。

③ 字符串

字符串是用单引号括起来的一串包括空格在内的ASCII字符。字符串中也能够含有单引号和反斜杠。

例：title ‘3:8encoder’；

‘It\’ s easy to use abel-hdl’；

‘He\\she can use abel-hdl’；

④ 注释

注释是为了使源文件易于理解，解释源代码本身不易理解的地方。它不是代码，对代码无影响。注释有下列两种形式。

(a) 用双引号开始，用另一个双引号或用换行结束。例如，“declaration section”。

(b) 用双斜杠开始，用换行结束。

⑤ 4种基数表示法

4种基数表示法如表2-9所示。

表2-9 4种基数表示法

基数名称	基数	前导符号	举例	十进制值
二进制	2	^b	^b101	5
八进制	8	^o	^o77	63
十进制	10	^d	^d43	43
十六进制	16	^h	^h0F	15

⑥ 专用常量

在赋值语句、真值表和测试向量中能够使用常量。可以首先将一个常量赋给标识符，然后在整个模块中使用这个标识符代表此常量。常量可以是数值，也可以是非数值型的专用常量值。

.C. 正脉冲时钟输入（低－高－低转换）

.K. 负脉冲时钟输入（高－低－高转换）

.D. 时钟下降沿

.U. 时钟上升沿

.F. 浮动输入或输出

.P. 寄存器预加载

.X.. 任意值

.Z. 高阻态测试输入或输出（三态值）

.SVn. n=2~9，驱动输入到过电压2~9

⑦ 运算符及其优先级

运算符及其优先级见表2-10。

表2-10　运算符及其优先级

类　别	运 算 符	优 先 级	说　明
算术运算符	−	1	取补
	<<	2	左移
	>>	2	右移
	*	2	乘法
	/	2	无符号除法
	%	2	取模
	+	3	加法
	−	3	减法
	!	1	非
	&	2	与
	#	3	或
	$	3	异或
	!$	3	同或
	==	4	等于
	! =	4	不等于
	<	4	小于
	<=	4	小于或等于
	>	4	大于
	>=	4	大于或等于

⑧ 赋值运算符

（a）= 和?=　非时钟赋值（组合逻辑输出）

（b）: = 和?:=　时钟赋值　（寄存器输出）

（c）在使用点扩展名时，使用=和?=运算符

（d）?=和?:=用于随意条件的赋值

例：q0 := a & b；　（没有指定寄存器类型）

　　q0.d = a * b；　（D型寄存器）

例：TRUTH_TABLE（[A，B]->F）；

　　[1，1]->0;

　　[1，0]->1;

　　[0，1]->1;

　　@DCSET

　　F=A&!B#!A&B;

　　F?=!A&!B

⑨ 集合

一个集合可以把若干信号和常量组合在一起，使用一个名字进行运算，简化了ABEL-HDL逻辑描述和测试向量。

（a）一组可作为整体进行运算的信号和常量

例：MULTOUT=[B0，B1，B2，B3，B4，B5，B6，B7]；

　　或MULTOUT=[B0.. B7]；

SELECT=[S0, S1, S2];
或SELECT=[S0..S2];

(b) 集合下标

使用集合下标能够访问集合内部的各个元素。

例：将一个16元素的集合中的4个元素赋值给一个较小的集合。

```
Declarations
   Set1=[f15..f0];
   Set2=[q3..q0];
Equations
   Set2:=Set1[7..4];
```

(c) 用集合运算实现布尔方程

例：ChipSel =A15&!A14&A13; (16位地址线中的高三位)
Addr=[A15, A14, A13]; (定义常量集合)
等价于
ChipSel=Addr==[1, 0, 1]; (方程)
或ChipSel=Addr==5;

(d) 集合运算（集合赋值以及集合和数值比较时应遵循的规则）

(i) 若该二进制数的有效位数多于集合中元素的个数，则从左边截去多余位；

(ii) 若该二进制数的有效位数少于集合中元素的个数，则从左边用0补齐。

例：[a, b]=^b101011; 等价于 [a, b]=^b11;
[a, b]=^b1; 等价于 [a, b]=^b01;

⑩ 块

(a) 块是用大括号括起来的文本段。用在逻辑等式、状态图、宏和命令中。块中的文本可以是一行，也可以是多行。

例：{This is a block}
{This is also a block, and it
spans more than one line.}

(b) 块可以嵌套。

例：{A=B&C;
{D=A; }
E=C;}

(c) 逻辑等式中使用块

例：
```
WHEN (Mode==S_Data) THEN Out_data:=S_in;
   ELSE WHEN (Mode==T_Data) THEN Out_data:=T_in;
WHEN (Mode==S_Data) THEN S_Valid:=1;
   ELSE WHEN (Mode==T_Data) THEN T_Valid:=1;
```
等价于
```
WHEN (Mode==S_Data) THEN
   {Out_data:=S_in; S_Valid:=1; }
```

```
ELSE WHEN (Mode==T_Data) THEN
{Out_data:=T_in; T_Valid:=1;}
```

(d) 状态图中使用块

例：
```
IF (Hold) THEN State1 WITH o1:=o1.fb; o2:=o2.fb;
                      ENDWITH
          ELSE State2;
```
等价于
```
IF (Hold) THEN State1 WITH {o1:=o1.fb; o2:=o2.fb;}
          ELSE State2;
```

例：
```
IF (Hold & !Reset)  THEN State1;
IF (Hold & Error)   THEN State2;
IF (!Hold )      THEN State3;
```
等价于
```
IF (Hold) THEN
{ IF (!Reset) THEN State1;
     IF (Error)  THEN State2;}
              ELSE State3;
```

⑪ 变量和变量置换

(a) 哑变量：在宏定义、模块或命令中被真实变量替代的标识符。

(b) 真实变量：用在宏定义、模块或命令中的变量，可替代哑变量。

(c) 在需要用真实变量替代哑变量的地方，哑变量前要加"?"，与其他标识符区分。

例：OR_EM MACRO (a, b, c) {?a # ?b # ?c}; （宏定义）

D=OR_EM (X, Y, Z) ; （宏引用）

ispEXPERT System文件后缀及其含义见表2-11。

表2-11 ispEXPERT System文件后缀及其含义

后 缀	文件类型	含 义
SYN	源文件	设计项目管理文件
ABL	源文件	ABEL硬件描述语言源文件
ABV	源文件	测试向量描述文件
SCH	源文件	电路原理图文件
VHD	源文件	VHDL硬件描述语言源文件
V	源文件	Verilog硬件描述语言源文件
PPN	源文件	引脚锁定描述文件（用电路图锁定引脚时为中间文件）
PAR	源文件	适配器控制参数文件
SYM	中间文件	电路符合文件
EQ0	中间文件	逻辑描述文件（由ABL编译所得）
EQ1	中间文件	简化逻辑文件（由EQ0化简所得）
EQ2	中间文件	带层次连接关系的逻辑描述文件
EQ3	中间文件	经优化的逻辑描述文件
EQ4	中间文件	经反复优化的逻辑描述文件
TMV	中间文件	经编译的测试向量文件

（续）

后 缀	文件类型	含 义
TT2	中间文件	逻辑网表输出文件，适配器输入文件
FXP	中间文件	逻辑布局结果文件
LST	中间文件	ABEL源文件的列表文件
LOG	中间文件	运行流程记录文件
SIM	中间文件	仿真用网表文件
JHD	中间文件	层次化关系连接表文件
JED	结果文件	熔丝图文件（JEDEC文件）
REP	结果文件	GAL器件设计编译报告文件
RPT	结果文件	IspLSI器件设计编译报告文件
XRF	结果文件	信号和节点简缩名称文件
ERR	结果文件	错误报告文件
MFR	结果文件	频率分析报告文件
TSU	结果文件	寄存器建立和保持时间报告文件
TPD	结果文件	TPD路径延时时间报告文件
TCO	结果文件	TCO路径延时时间报告文件

后面将进行在系统编程设计训练实验。

实验一　格雷码计数器

一、实验目的

（1）以GAL16V8为例，了解GAL的工作原理、特点、引脚和使用方法。

（2）初步掌握使用ABEL语言编程的方法，并且使用ABEL语言编程实现较复杂的逻辑功能。

（3）初步了解Synario软件的使用，学会写GAL的方法。

二、实验设备与器材

（1）GAL16V81片。

（2）GAL编程器（TDS系统选件）1套。

（3）示波器1台。

三、实验内容及实验步骤

（1）用ABEL编程语言设计一个4位格雷码计数器。格雷码的编码规则规定，任何相邻的代码只有一个二进制位状态不同，其余三位二进制位必须状态相同。用4个二进制位的格雷码表示十进制数的10个状态的方案很多，表2-12给出两组最常用的编码值。本实验只要求实现一种格雷码方案，从两个方案中任选其一。

表2-12　格雷码

十进制数	格雷码（1）	格雷码（2）	十进制数	格雷码（1）	格雷码（2）
0	0000	0000	5	1110	1011
1	0001	0100	6	1010	0011
2	0011	0110	7	1000	0001
3	0010	0010	8	1100	1001
4	0110	1010	9	0100	1000

（2）运行Synario软件，键入格雷码计数器的设计方案。并进行编译和连接，生成JEDEC类型（扩展名为JED）的文件。

（3）实验内容（2）也可用FM软件实现。首先用文本编辑生成一个扩展名为PLD的设计文件，该设计文件应符合FM规定的语法。然后用FM软件编译设计文件，生成JEDEC类型的文件。

（4）将JEDEC文件通过GAL编程器写入GAL16V8。

（5）将GAL16V8插入TDS实验台上的20芯插座，接好电源和地。使用单脉冲做计数时钟。格雷码计数器的4个输出接电平指示灯。观察并记录单脉冲多次作用下，格雷码输出的状态变化。

四、实验报告要求

（1）写出实现格雷码计数器的ABEL程序。

（2）根据自己的设计，比较用语言设计数字逻辑电路和用原理图设计数字逻辑电路的优缺点。

（3）根据自己的设计，简述使用可编程逻辑器件（例如GAL）实现数字逻辑功能有哪些特点。

实验二　3-8线译码器

一、实验目的

（1）学习使用ABEL语言中的逻辑式法设计简单组合逻辑电路。

（2）学习使用ABEL语言中的真值表法设计简单组合逻辑电路。

（3）学习使用ispEXPERT软件。

（4）学习ISP器件的编程和下载。

二、实验设备与器材

（1）ISP10161片。

（2）示波器1台。

（3）数字电路实验系统1台。

三、实验内容及实验步骤

（1）用ABEL语言中的真值表法和逻辑等式法设计一个3-8线译码器。3-8线译码器的真值表如表2-13所示。

（2）在ispEXPERT环境下，输入两种设计方案，并进行编译、连接、器件适配和下载。

（3）在实验台上验证设计的正确性。接好电源和地。4个输入接实验台的电平开关输出，译码器的8个输出接电平指示灯。改变器件的输入电平，观测并记录指示灯的显示。

表2-13　3-8线译码器的真值表

输入				输出							
EN	IN2	IN1	IN0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	1	1	0	1	1

（续）

输入				输出							
EN	IN2	IN1	IN0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
1	0	1	1	1	1	1	1	0	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1

四、实验报告要求

（1）写出用真值表法设计3-8线译码器的ABEL语言方案。

（2）写出用逻辑式法设计3-8线译码器的ABEL语言方案。

（3）写出用真值表法和逻辑式法设计组合逻辑电路的心得体会。

实验三　8421BCD码计数器

一、实验目的

（1）学习使用ABEL语言中的逻辑式法设计简单组合逻辑电路。

（2）学习使用ABEL语言中的真值表法设计简单组合逻辑电路。

（3）学习使用ispEXPERT软件。

（4）学习ISP器件的编程和下载。

二、实验设备与器材

（1）ISP10161片。

（2）数字电路实验系统1台。

（3）示波器1台。

三、实验内容及实验步骤

（1）用ABEL语言中的真值表法和逻辑式法设计一个十进制计数器。该计数器的时钟输入为CLK，在CLK的上升沿计数。它具有同步复位输入端CLR，当CLR为低电平时，在下一个时钟的上升沿将计数器清零。该计数器的计数顺序如下。

（2）在ispEXPERT环境下，输入两种设计方案，并进行编译、连接、器件适配和下载。

（3）在实验台上验证设计的正确性。接好电源和地。使用单脉冲作计数器的时钟脉冲，复位信号接电平开关。计数器的输出Q3、Q2、Q1、Q0接电平指示灯。8421BCD码计数器的真值表如表2-14所示。

表2-14　8421BCD码计数器真值表

计数	输出			
	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1

（续）

计数	输出			
	Q3	Q2	Q1	Q0
6	0	1	1	0
7	0	1	1	0
8	1	0	0	0
9	1	0	0	1

四、实验报告要求

(1) 写出用真值表法设计十进制计数器的ABEL语言方案。

(2) 写出用逻辑式法设计十进制计数器的ABEL语言方案。

(3) 写出用真值表法和逻辑式法设计组合逻辑电路的心得体会。

实验四　简单电子琴

一、实验目的

(1) 掌握较复杂逻辑电路的设计和调试。

(2) 进一步掌握用ABEL语言设计数字逻辑电路。

(3) 熟悉ispEXPERT软件的使用方法。

(4) 熟悉ISP器件的使用。

(5) 了解音调的初步知识。

二、实验设备与器材

(1) PLCC封装的ISP1016或者M4-64/32一片。

(2) 示波器1台。

(3) 万用表1块。

三、实验内容

(1) 用ABEL语言设计一个电子琴。使用TDS实验台上的8个电平开关做琴键。电平开关输出为高电平时相当于琴键按下，电平开关输出为低电平时相当于琴键松开。电子琴共有C调的8个音：1、2、3、4、5、6、7和$\dot{1}$。

(2) 在ispEXPERT中，将设计好的程序输入、编译和连接，生成JEDEC格式的文件。

(3) 将JEDEC格式的文件下载到器件中。

(4) 在TDS数字电路实验台上对设计进行调试。调试时用实验台上的小喇叭做发声装置。

四、实验提示

(1) C调的音符与频率的关系如表2-15所示。

表2-15　音符与频率的关系

音符	1	2	3	4	5	6	7	$\dot{1}$
频率（Hz）	262	294	330	349	392	440	494	523

只要向TDS实验台上的喇叭输出某一频率的方波，喇叭就发出相应音调的声音。将实验台喇叭区域的开关J1置为开路，从“输入”插孔向驱动喇叭的三极管基极送控制信号，则控制喇叭按希望的频率发声。

(2) 设计一个多模计数器，对实验台上的某一时钟（例如100kHz）进行分频，产生8种希望的频率。注意驱动喇叭的方波占空比应是50%，以增大音量。

(3) 根据开关电平输出确定9种状态（包括不发声状态）之间的转换。

(4) ISP1016或者M4-64/32的引脚图见附录1。

五、实验步骤（以Lattice公司ispLSI1016-60 PLCC44芯片为例）

1. 设计思想

本项目由两个模块组成。

顶层模块piano 主要由一个多模计数器和二分频计数器组成，二分频计数器输出送往喇叭驱动级，控制喇叭发出相应音乐。为了提高音量，输出信号的占空比为50%，由于多模计数器输出占空比不能满足要求，因此多模计数器的输出又接了一级占空比为50%的二分频计数器。在某一音频下，多模计数器的模应为

$$M = 100\,000/2/f$$

其中，M代表计数器的模，f代表音频的频率。例如，当按下"2"键时，为使二分频计数器输出频率为294Hz的方波，计数器的模应为100 000/（294*2）=170。

底层模块value 用来判断当前按键状态是否有效。所谓按键状态有效是指，有一个键按下且同一时刻只有一个键按下。本实验中用电平开关代表按键，即任一时刻只能有一个开关置为1状态。按键状态有效时，right = 1；按键状态无效时，right = 0。

2. 新建项目

打开ispEXPERT软件，选择File菜单中的New Project，弹出"Create New Project"对话框。在对话框的Directories一栏中，选择一个目录（最好是为该项目专门建立的目录），存放该项目中的所有文件。然后在对话框的Project File Name一栏中输入项目文件名piano.syn，并按"确定"键关闭对话框。

3. 选择器件

双击Sources in Project 窗口中Virtual Device项，弹出"Choose Device"对话框，选择器件ispLSI1016-60 PLCC44。

4. 建立源文件

选择Source菜单中的 New，选择ABEL-HDL Module，按OK键。在弹出对话框的Module Name栏中输入模块名，按OK键，进入ispEXPERT Text Editor，开始输入源文件。本项目由顶层模块piano和底层模块value组成。

5. 顶层模块piano的ABEL语言源文件

```
MODULE piano

DECLARATIONS
    // lower module declaration
value       interface  (d1..d7, di -> right) ;
value_0     functional_blockvalue;

    // input
clock              pin 11;                 // 时钟输入 100kHz
d1, d2, d3, d4      pin 9, 40, 36, 3;      // 琴键
d5, d6, d7, di      pin 29, 10, 16, 43;
```

```
    // output
mu                  pin 38 istype 'reg' ;    // 音频输出
    // node
mu2                 node;
mu0, mu1            node;
q0..q7              node istype 'reg' ;      // 多模计数器
q = [q7..q0];
d = [di, d7..d1];
EQUATIONS
value_0.[di, d7..d1] = d;
    // 多模计数器，模191、模170、模151、模143、模128、模114、模101和模97
q.clk = clock;
q := (q+1) & !mu2 & value_0.right;
mu0 = (q == 190) & (d1 == 1) # (q == 169) & (d2 == 1) # (q == 150) & (d3 == 1)
        # (q == 142) & (d4 == 1) ;
mu1 = (q == 127) & (d5 == 1) # (q == 113) & (d6 == 1) # (q == 100) & (d7 == 1)
        # (q == 96) & (di == 1) ;
mu2 = mu0 # mu1;
mu.clk = mu2;                                // 二分频计数器
mu := !mu;
END
```

6. 底层模块value的ABEL语言源文件

```
MODULE value
DECLARATIONS
        // input
d1..d7, di          pin;          // 琴键1, 2, 3, 4, 5, 6, 7和i
        // output
right               pin;          // 为1 表示当前按键有效
        // node
right0, right1      node;
d=[d1..d7, di];
EQUATIONS
right0 = (d == ^b00000001) # (d == ^b00000010) # (d == ^b00000100)
       # (d == ^b00001000) ;
right1 = (d == ^b00010000) # (d == ^b00100000) # d == ^b01000000)
       # (d == ^b10000000) ;
right = right0 # right1;          // 为1表示当前按键有效
END
```

7. 编译源文件

选择Source in Project窗口中的ABEL语言源文件，双击Process For Current Source窗口中的Compile Logic和Reduce Logic。如果在两命令前出现对号，则表示编译通过，否则根据错误提示修改源文件。

8. 生成JED文件

选择Source in Project 窗口中的 ispLSI1016-60 PLCC44，依次执行Process For Current Source窗口中的Link Design和Fit Design。如果在JEDEC File前出现对号，则表示piano.jed文件已生成，否则根据错误提示进行修改。

9. 下载JED文件到ispLSI1016芯片

用Lattice编程电缆将TDS数字电路实验台与主机连接，将ispLSI1016芯片插入对应PLCC插座，接通电源。选择Source in Project窗口中的ispLSI1016-60 PLCC44，双击Process For Current Source窗口中的isp Download System，则打开ISP Daisy Download软件。执行Configration菜单中的Scan Board，点击窗口中的browse键，选择piano.jed。选择Operation中的Program & Verify，执行Command菜单中的Run Operation，有关信息在Messages窗口中显示。

10. 调试片内程序

根据piano.abl文件中的引脚定义连线，即，ispLSI1016芯片的引脚11接实验台100kHz时钟输出端，引脚38接实验台喇叭“输入”端，引脚9、40、36、3、29、10、16、43分别接实验台的逻辑开关输出K1、K2、K3、K4、K5、K6、K7、K8。K1、K2、K3、K4、K5、K6、K7、K8分别代表音符1、2、3、4、5、6、7、$\dot{1}$。当逻辑开关输出只有一个为高电平时，喇叭应发出相应的音乐。

2.3 课程设计选题

课程设计一 出租车计价器

1. 设计说明

出租车计价器是我们常见的一种设备，它可以用在系统可编程技术来实现。本课程设计的任务就是设计一个出租车计价器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 根据出租车上的速度传感器传来的脉冲个数（对应里程）和设置的里程单价来计算对应的总价格，并将总价格通过LED实时显示。

(b) 起步价可以设置。

(c) 里程单价可以设置。

(d) 可以对总价格进行复位，从而为下次计费作好准备。

4. 设计提示

本设计应主要实现两个功能，显示行驶里程与总价。行驶里程的计算可由计数器实现，每一个脉冲假设为1km。实验系统有标准时钟信号，因此通过分频可以获得脉冲信号（对应里程）。

总价则可由可控制条件的累加器实现，然后予以输出。累加器可以由一个加法器和一个可复位的寄存器组成，最后设计译码器，将2位的十六进制转化为BCD码。

课程设计二　自动打铃机控制器

1. 设计说明

自动打铃控制器是根据学校冬季或夏季的作息时间，对广播、电铃和电灯进行自动控制并具有校时功能的设备。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 可以选择冬季或夏季作息时间。

(b) 自动打铃控制器可以控制电铃、电灯和广播，因此必须有3个输出信号来控制电铃、电灯和广播。

(c) 校时功能可通过对输入端的触发来实现，每天中午12点准时校时，当北京时间12点整时，给输入端一个触发，使控制器自动复位到12点。

4. 设计提示

本设计可由一个控制器模块完成所有的功能，控制器框图如图2-34所示。

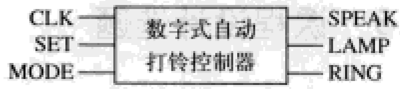


图2-34　控制器框图

图中，

CLK——外部时钟脉冲输入，频率为1Hz。

SET——校时，高电平有效。

MODE——模式选择，“1”为夏季作息时间，“0”为冬季作息时间。

SPEAK——输出信号，控制广播。

LAMP——输出信号，控制电灯。

RING——输出信号，控制电铃。

该控制器需要一个内部计数器来计时，共需计24个小时，计时精确到秒。

通过对内部节点min和hour的检测，可以定时触发广播、电铃和电灯。

打铃时间长度由外部时钟信号CLK控制。

课程设计三　复印机控制电路

1. 设计说明

复印机在复印工作中需要计数器的控制，它可以用数字电路来实现。本课程设计的任务就是利用在系统可编程技术或者中规模逻辑器件设计一个复印机控制电路。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）或进行电路原理设计，画出电路原理图。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 设置复印数。从键盘（0~9）输入复印的张数。

(b) 显示复印数。显示位数为3位（或2位），最大数为999（或99）。

(c) 按运行（RUN）键后，电路就自动工作。

(d) 显示的复印数能自动递减。每复印一张，数字递减一次，直到0就停机。

4. 设计提示

本设计需要注意以下3个问题。

(a) 置数：由于实验系统中没有提供题目要求的键盘，所以用二进制开关手动置数。

(b) 分频：通过分频由实验系统中标准时钟信号得到1Hz的脉冲信号。

(c) 实现模1000（或100）的减法计数器，并将数据显示在LED上。

课程设计四　自动洗衣机控制器

1. 设计说明

本课程设计的任务是设计一个模拟洗衣全过程的自动洗衣机控制器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

假设自动洗衣机的定时操作顺序是，洗衣10min、排水2min、脱水3min，然后停止。设计出这个自动洗衣机的控制器。

4. 设计提示

本设计有4个状态，分别为初始状态、洗衣状态、排水状态和脱水状态。当有复位信号时，系统进入循环控制状态，依次执行操作，可从信号灯观察到所处状态。注意：当中途断电或置复位后，系统不会自动关闭，而是继续执行未完的程序，直到一个循环结束才终止。

课程设计五　数字密码锁电路

1. 设计说明

本课程设计的任务是设计一个数字密码锁。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 设14位数字码分别为高7位数码和低7位数码，用数字逻辑开关预置，输出信号为OUT。OUT为1时，表示锁开启。

(b) 14位数字码分时操作，先预置高7位数码，然后置入低7位数码。

(c) 要求电路工作可靠，保密性强，开锁出错立即报警（用声音报警）。

4. 设计提示

本设计包括密码预置、密码验证和密码修改3个部分。

密码预置时，采用实验系统中的7个逻辑开关分时进行高7位和低7位密码输入，可通过一个电平开关来控制分时操作。

密码验证可通过时钟信号控制输入密码与预置密码的比较。相同则绿灯亮，不同红灯亮并且报警。

密码修改类似密码预置，通过脉冲信号确认修改值。

课程设计六　简易频率计

1. 设计说明

频率计是用来测量信号频率的电子设备。其工作原理是，首先产生一系列准确的闸门信号，例如1ms、0.1s和1s等，然后用这些闸门信号控制一个计数器对被测脉冲进行计数，最后将结果显示出来。如果闸门信号是1s，那么1s内计数的结果就是被测信号的频率。如果闸门信号是1ms，那么计数结果是被测信号频率的千分之一，或者说结果是以kHz为单位的频率值。

本课程设计的任务就是设计一个用于测量1MHz以下数字脉冲信号频率的简易频率计，即6位数字频率计。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

用于测量1MHz以下数字脉冲信号的频率。闸门只有1s一档。测量结果在数码管上显示出来。

4. 设计提示

(a) 频率计中，最原始的时基信号准确度一定要高。可用100kHz时钟信号做原始时基信号。

(b) 1s的闸门信号是由100kHz时钟经5次10分频后，再经2分频产生。这样产生的闸门信号脉宽是1s，占空比是50%。在2s的时间内，1s用于计数，1s用于显示结果。

(c) 用于被测信号计数的计数器应采用十进制。测得的结果可直接送6个数码管显示。每次对被测信号计数前，计数器应清零。

课程设计七　交通控制灯

1. 设计说明

十字路口的红绿灯非常普遍。本课程设计的任务是设计一个方便交通管理的交通控制灯。

2. 设计内容：

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

以4个红色指示灯、4个绿色指示灯和4个黄色指示灯模仿路口的东、西、南、北4个方向的红、绿、黄交通灯。控制这些指示灯，使它们按下列规律亮和灭。

(a) 初始状态为4个方向的红灯全亮，时间1s。

(b) 东、西方向绿灯亮，南、北方向红灯亮。东、西方向通车，时间5s。

(c) 东、西方向黄灯闪烁，南、北方向红灯亮，时间2s。

(d) 东、西方向红灯亮，南、北方向绿灯亮。南、北方向通车，时间5s。

(e) 东、西方向红灯亮，南、北方向黄灯闪烁，时间2s。

(f) 返回（b），继续运行。

(g) 若发生紧急事件，例如救护车和警车通过时，则按下单脉冲按钮，使得东、西、南、北4个方向的红灯全亮。紧急事件结束后，恢复到原状态继续运行。

4. 设计提示

(a) 一个典型的时序状态机，一共有6个大的状态。

(b) 黄灯闪烁可通过连续亮0.2s和灭0.2s实现。

(c) 利用实验系统中1kHz作为设计中的初始时钟，通过分频得到0.2s、1s和5s等时钟信号。

(d) 紧急事件发生时，要注意保存必要的信息，以备紧急事件结束后恢复到原状态继续运行。

课程设计八　电子钟

1. 设计说明

本课程设计的任务是设计一个具有显示时间、复位、启动及在停止状态下修改时间并能整点报时等功能的电子钟。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 6个数码管显示时、分、秒。

(b) 能使电子钟复位（清零）。

(c) 能启动和停止电子钟运行。

(d) 在电子钟停止运行状态下，能够修改时、分、秒的值。

(e) 具有报时功能，整点时喇叭鸣叫。

4. 设计提示

本设计采用同步时序逻辑电路的设计方法，具体可由一个顶层模块和三个底层模块来实现。顶层模块负责的是秒脉冲的产生、所有输入输出功能的实现以及各子模块之间的协调，

可用原理图方法实现。三个底层模块分别实现的是时、分、秒计数器的功能，可用ABEL-HDL语言设计。为获得秒脉冲信号，需对实验系统提供的脉冲进行分频。

课程设计九　拔河游戏机

1. 设计说明

本课程设计的任务是利用在系统可编程技术设计一个模拟拔河运动的电路。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 拔河游戏机需要9个发光二极管排成一行，开机后只有中间一个亮点，作为拔河的中间线。游戏双方各持一个按键，迅速且不断地按动产生脉冲，哪方按得快，亮点就向哪方移动，每按一次，亮点移动一次。移到任一方二极管的终端，该方就获胜。此时双方按键均无作用，输出保持，只有经复位后才能使亮点恢复到中心线。

(b) 显示器显示胜者胜利的次数。

4. 设计提示

按键信号是输入的脉冲信号，每按一次按键都应能进行有效的计数。建议采用可逆计数器的加、减计数输入端，分别接受两路脉冲信号，经译码器输出，使中间的二极管发亮。在加减计数过程中，注意亮点的移动。当亮点移到任一方终端时，产生一个信号，使计数器停止计数。两方的胜利都分别进行计数器计数。注意设置复位信号，以使游戏回到初始状态。

课程设计十　简易投币式自动售货机

1. 设计说明

本课程设计的任务是设计一个简易投币式自动售货机。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

自动售货机可以销售许多种类的商品，通过选择需要的商品按钮，投入一定数量的纸币，顾客就可以获得所需的商品。该自动售货机应具有如下功能。

(a) 设自动售货机能销售3种商品，热狗、汉堡和双层汉堡，它们的单价分别是1元、4元和8元，并且机器中3种商品的数量无限。

(b) 自动售货机允许投入1元、2元和5元纸币。当总投入的币值等于顾客需要的商品单价时，机器送出需要的商品；若总投入的币值大于顾客需要的商品单价时，机器除提供需要的商品之外，还要将余币退出；若总投入的币值小于顾客需要的商品单价时，则机器退出顾客投入的纸币。另外，设机器内1元和2元找零纸币无限。

(c) 如果投入的纸币达到或大于所要购买商品的价格就不必再投入纸币了，自动售货机会发出一个指示信号使接受纸币的装置不再接收更多的纸币。

(d) 为提高自动售货机的效率，每次投币时间应有规定。设定每次购买商品的允许投币时间为30s，在此时间到的时候，总投币值不足顾客购买的商品单价时，售货机按不足钱数处理，退还全部投入纸币。在设定时间内，总投币值不足顾客购买的商品单价时，若需取消交易则可按取消键，售货机按不足钱数处理，退还全部投入纸币。

(e) 当顾客完成一次购买之后或按错所需要的商品按钮，需要设置一个重新开始按钮，使整个系统恢复到初始状态；

(f) 顾客一次只能购买一个商品，若需要更多的商品，则需要重复第一次的操作。

4. 设计提示

本设计实际是一个迷你型同步时序逻辑设计，关键在于将实际模型转化为状态转换图。

设计可分为30s计时部分、投钱计数部分以及比较部分。需考虑初始情况，投入为1元、2元和5元的情况，当投入的钱达到所购买的商品价格时，就输出相应商品且跳回初始状态。

课程设计十一　任意进制加法计数器

1. 设计说明

任意进制加法计数器通过改变计数模式来实现多种计数方式。广泛应用于计数、分频和定时等场合。本课程设计的任务是利用在系统可编程技术设计一个任意进制加法计数器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 加法计数器的进制从二进制到十六进制可变；

(b) Q0~Q3是输出，Q3为高位；CLK是计数脉冲输入；EN是使能端，低电平有效；进制由M0~M3控制，当取值为0000时，计数器复位；取值为0001时，计数器为二进制计数；取值为0010时，为三进制计数，依次类推。

4. 设计提示

适当选择输入方式和中间节点，将有助于设计的完成。

课程设计十二　多功能十进制计数器

1. 设计说明

多功能十进制计数器既可做加法计数器，又可做减法计数器，是一种常用的计数器。本课程设计的任务是利用在系统可编程技术设计一个多功能十进制计数器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) EN是使能端，低电平有效；Q0~Q3是输出，Q3为高位。

(b) 有两个控制输入信号C1和C0。

- 当C1C0=00时，计数器复位。
- 当C1C0=01时，计数器预置一个十进制数。
- 当C1C0=10时，减法计数。
- 当C1C0=11时，加法计数。

4. 设计提示

计数器的10个状态为S0=1111，S1=1110，S2=1101，…，S8=0111，S9=0110。

课程设计十三 1011序列发生器和检测器的设计与实现

1. 设计说明

本课程设计的任务是设计一个1011序列发生器和检测器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 设计一个1011序列发生器。

(b) 设计一个1011序列检测器，该序列检测器的输入可以通过人工拨动开关来选择（被检测的序列可以通过人工拨动开关产生，也可以将序列发生器的输出作为序列检测器的输入）。

4. 设计提示

本设计的关键是画出正确的1011序列发生器和检测器的状态转换图。建议用ABEL语言中的状态图描述语句进行设计。

课程设计十四 铁路和公路交叉路口交通控制电路

1. 设计说明

本课程设计的任务是设计一个铁路和公路交叉路口的栅门的控制电路。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

交叉路口的示意图如图2-35所示，图中P1、P2为两个压敏传感器，P1与P2之间的距离大于火车的长度，A、B为两个栅门。

当火车任何部分处于P1和P2之间时，栅门应同时关闭，否则同时打开。P1、P2的输出分别用X1、X2表示，X1、X2控制一数字电路，其输出为Z，用Z来操纵A、B。

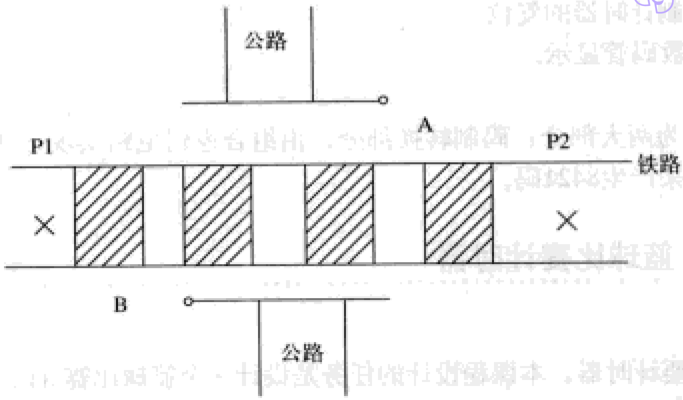


图2-35 交叉路口示意图

4. 设计提示

(a) 压敏元件的作用是，当它承受火车的压力时，产生逻辑电平“1”，否则为“0”。

(b) 由控制电路Z控制栅门：Z=1时，栅门A和B关闭；Z=0时，栅门A和B打开。

(c) 火车通过交叉路口的运行过程及在此过程中控制电路的输入、输出情况如下所示。

① 火车尚未到来：X1X2=00，Z=0。

② 火车由西向东行驶压在P1时：X1X2=10，Z=1。

③ 火车继续由西向东行驶且位于P1与P2之间时：X1X2=00，Z=1。

④ 火车继续由西向东行驶压在P2时：X1X2=01，Z=1。

⑤ 火车由东向西行驶压在P2时：X1X2=01，Z=1。

⑥ 火车继续由东向西行驶且位于P1与P2之间时：X1X2=00，Z=1。

⑦ 火车由东向西行驶压在P1时：X1X2=10，Z=1。

注：火车由西向东行驶离开P2以后和火车由东向西行驶离开P1以后，情况皆与①相同。

根据分析可以确定电路的状态，然后化简状态，分配状态，最后通过编程实现功能。

课程设计十五 码制转换器的设计与实现

1. 设计说明

本课程设计的任务是设计一个码制转换器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 设计一个双向转换电路，完成8421码和Gray码的互换。同时设计一个4位二进制计数器产生8421码进行测试。

(b) 人工拨动开关产生的时钟可以完成计数的增加，并且把计数的结果作为码制转换的输入直接进行转换。

(c) 拨动开关来选择是哪一种码制转换（8421码转为Gray码，还是Gray码转为8421码）。

(d) 用开关控制计时器的复位。

(e) 可以通过数码管显示。

4. 设计提示

本设计可以分为两大部分：码制转换部分，由组合逻辑电路实现；计数器部分，为4位二进制计数器，用来产生8421码。

课程设计十六　篮球比赛计时器

1. 设计说明

篮球比赛中需要计时器，本课程设计的任务是设计一个篮球比赛计时器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 篮球比赛上下半场各20min，要求能随时暂停，启动后继续计时，一场比赛结束后应可清零重新开始比赛。

(b) 计时器由分、秒计数器完成，秒计数器为模60，分计数器应能计至40min。

(c) “分”、“秒”显示用LED数码管，应配用相应译码器。

(d) 用人工拨动开关来控制计时器的启动/暂停。

(e) 半场、全场到时会自动有相应的提示。

4. 设计提示

本设计可由计数器、译码显示电路、启/停控制电路、半/全场时间到音响提示电路等组成。设计中需要模60的计数器和分频器。控制电路要注意控制计时器的清零和暂停。通过判断计数值来提示半场和全场到。

课程设计十七　自动电梯控制系统

1. 设计说明

本课程设计的任务是设计一个自动电梯控制系统。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）或进行电路原理设计，画出电路原理图。

(c) 对设计电路进行模拟与测试。

3. 设计要求

为一幢9层的高楼设计一个自动控制电梯。

(a) 系统控制的电梯往返于1~9层楼。

(b) 乘客要去的楼层数可手动输入并显示（设为A数）。

(c) 电梯运行的楼层数可自动显示（设为B数）。

(d) 当A>B时，系统能输出使三相电机正转的时序信号，使电梯上升；
当A<B时，系统能输出使三相电机反转的时序信号，使电梯下降；
当A=B时，系统能输出使三相电机停机的信号，使电梯停止运行并开门。

(e) 电梯是上升还是下降，各层电梯门外应有指示，各层电梯门外应有使电梯上升或下降到乘客所在楼层的控制开关。

4. 设计提示

本设计可以通过比较器对当前楼层数和乘客所要去的楼层数进行比较，产生的输出信号控制加减计数器，模拟电梯的上升和下降。

电梯运行速度可以由实验系统标准时钟信号通过软件分频获得的1Hz脉冲信号进行控制。

乘客要去的楼层数可由编码器实现。

课程设计十八　篮球30s定时控制电路设计

1. 设计说明

定时器的使用场合非常广泛，在篮球比赛中，30s定时器是必备的电子设备。本课程设计的任务是设计一个篮球30s定时控制电路。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（ABEL-HDL源程序或原理图）。

(c) 对设计电路进行模拟与测试。

3. 设计要求

(a) 有30s计时功能。

(b) 设置外部操作开关，具有对计时器清零、启动和暂停/连续等功能。

(c) 计时器为30s递减计时器，其计时时间间隔为1s。

(d) 计时器递减计时到零时，数码显示器不能灭灯，同时发出报警信号。

4. 设计提示

本设计可由秒发生器、外部控制、显示和报警4个部分组成。秒发生器通过实验系统标准时钟信号分频产生1s的时钟；外部控制实现置数、暂停和清零等功能；显示通过实验系统提供的数码管完成；报警电路由报警条件触发。

课程设计十九　节日彩灯控制器

1. 设计说明

节日彩灯的变化可以烘托节日的欢乐气氛，彩灯的变化需要用控制器进行控制，复杂的控制器可能需要微处理实现，但简单的彩灯控制器用基本的数字电路即可以实现。本课程设计是要设计一个简单的控制电路，用来控制8个彩灯按事先设计好的变化规律周期性地变化。

2. 设计内容

(a) 进行需求分析和信号定义（分析控制器的基本功能和要求，对信号及属性进行定义）。

(b) 设计节日彩灯变化规律。

(c) 设计和分析电路原理（给出电路原理图、电路参数及计算过程）。

(d) 设计测试方法和测试数据，安装电路，并进行实验调试。

3. 设计要求

(a) 灯的变化要求符合人的视觉习惯，体现出节日的欢乐气氛，富有变化，具有周期性，彩灯数量不少于8个。

(b) 通过计算确定电路参数，如变化周期、时钟频率等。

(c) 能进行启动、暂停和复位控制。

(d) 画出电路原理图，在实验箱上进行电路调试。

(e) 完成课程设计报告。

4. 设计提示

图2-36中时钟发生电路产生1Hz的时钟，用于控制彩灯的变化节奏。*N*进制状态计数器对时钟进行计数，产生并记忆彩灯变化的*N*个状态，并控制状态循环。译码电路对*N*个状态译码产生彩灯的控制信号，经驱动电路放大后加载于彩灯。实验系统有标准的1kHz时钟信号，也可以通过分频电路获得所需时钟信号。



图2-36 节日彩灯控制电路框图

课程设计二十 智力竞赛电子抢答器

1. 设计说明

在一些智力游戏活动中，通常用电子抢答器来决定谁进行本轮答题，简单的抢答器可以用数字电路来设计。本课程设计的任务是设计一个电子抢答控制器决定最先给出控制信号的答题人。

2. 设计内容

(a) 进行需求分析和信号定义。

(b) 设计、分析电路原理，画出电路原理图。

(c) 安装电路并调试。

3. 设计要求

(a) 通道数不少于4个，输入用非自锁的开关键表示，输出用LED指示。

(b) 只对最先给出控制信号的通道做出反映，对应的输出灯亮有效。

(c) 设置主持人清除信号，有效时清除输出并使电路复位到初始状态。

(d) 输出信号要求锁存和保持直到清除信号有效时为止。

(e) 按要求完成课程设计报告。

4. 设计提示

电子抢答器本质上就是优先级编码器。电路可以由两部分构成，优先级编码器和输出驱动显示电路。有两种方法得到优先级编码器，一是直接利用优先级编码器IC，二是利用可编程逻辑电路定制。优先级编码器设计的关键是，一旦检测到一个通道信号有效，立即封锁其他通道的信号输入，并对有效通道进行编码、锁存和输出。本电子抢答器的框图如图2-37所示。



图2-37 智力竞赛电子抢答器的电路原理框图

课程设计二十一 数字PWM调节器

1. 设计说明

PWM即脉宽调制，PWM技术广泛应用在开关电源、电机调速和无线遥控中。PWM有模拟和数字两种。数字PWM就是用写入的数字控制调节输出脉冲的宽度。本课程设计的任务是设计数字PWM调节器电路。

2. 设计内容

(a) 进行调节器需求分析，定义信号及属性。

(b) 设计、分析电路原理，计算电路参数，画出电路原理图。

(c) 安装电路，并调试。

3. 设计要求

(a) 控制数据宽度为8bit，以串行或并行输入，要求能锁存。

(b) 输出信号为方波，占空比为1：2，电平为TTL电平。

(c) 设置复位和使能信号。复位信号使电路复位到初始状态，使能信号使电路输出有效。

(d) 按要求完成设计报告。

4. 设计提示

PWM电路可以由时钟电路、数据寄存器和分频电路组成，如图2-38所示。时钟电路提供时钟信号给分频电路，同时时钟和写入的控制数据共同决定PWM脉宽调节精度。数据锁存器用于接收和存放调节控制数据，用它控制分频电路的分频系数。

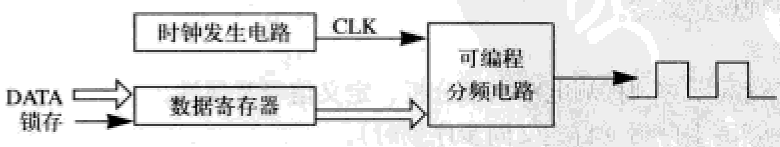


图2-38 PWM调节器电路原理框图

课程设计二十二 汽车尾灯控制器

1. 设计说明

汽车通常用尾灯来指示当前的运行状态，如左转弯、右转弯、紧急停车、夜间行使、夜

间停车和安全停车等。有限个状态可用若干个灯的组合表示，通常是右转弯时右灯闪烁，左转弯时左灯闪烁，紧急停车左右灯同时闪烁，夜间行使车时左右小红灯常亮，安全停车所有灯全熄灭，本课程设计的任务是完成汽车尾灯控制器的设计。

2. 设计内容

(a) 进行控制器需求分析，定义信号及属性。

(b) 设计灯位置、数量及变化规律。

(c) 设计和分析电路原理，画出电路原理图。

(d) 安装电路，并调试。

3. 设计要求

(a) 能反映出左转弯、右转弯、夜间行车、紧急停车、夜间停车和安全停车等状态，可以自己扩展。

(b) 灯的闪烁频率的设置符合人们视觉习惯，1~2Hz左右。

(c) 按要求完成实验设计报告。

4. 设计提示

图2-39中时钟发生电路产生1Hz的时钟，用于控制灯的闪烁频率。*N*进制状态计数器对时钟进行计数产生并记忆*N*个状态，每个计数值仅对应一个工作状态，相当于状态编码，通过对状态编码译码获得所需的控制信号，经驱动电路放大后控制尾灯变化。

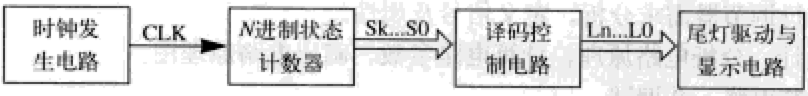


图2-39　汽车尾灯控制电路框图

课程设计二十三　帧同步序列检测电路

1. 设计说明

在数据通信系统中，链路层上信息以帧为单位进行通信。帧是一个信息结构，要求每帧的开头用一个特定的序列标识，用在帧的开始和结束位置，如HDLC规程的帧标志序列为01111110，所以接收端为了准确地提取一个帧的有用信息，首先要从连续的串行码流中检测出帧同步序列。帧同步序列检测可以用硬件电路实现，本课程设计的任务是设计一个01111110的序列检测电路。

2. 设计内容

(a) 进行帧同步序列检测电路需求分析，定义信号及属性。

(b) 定义帧同步序列（自定义同步序列时）。

(c) 设计和分析电路原理，画出电路原理图。

(d) 设计测试方法和测试数据，安装电路，并调试。

3. 设计要求

(a) 信息串行输入，能分辨连续的帧同步序列。

(b) 用LED指示检测结果，检测到一个有效标志时输出指示灯闪烁一次。

(c) 同步码长度规定为8bit，可考虑同步码的修改（内容和长度可变）。

(d) 按要求完成课程设计报告（电子文档和打印或手写各一份）。

4. 设计提示

当要求同步码可修改时可以采用寄存器先锁存同步码，即电路由两部分组成，帧同步码锁存电路和检测电路。电路框图如图2-40所示。

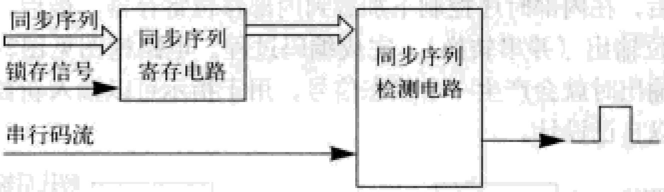


图2-40　帧同步序列检测电路框图

课程设计二十四　4b/5b编解码器

1. 设计说明

在数字通信系统中，为了拟制数字信号直流分量和方便接收器提取同步时钟，需要对数字信号进行扰码，扰乱数字码流中的连0连1情况，使0和1的游程满足要求，其中有一种简易的扰码方法，就是把4bit编码成5bit，即插入一位冗余比特，使5bit编码中的1bit与前4bit中的一位成反码关系，这样就破坏了连0连1的情况降低了直流成分，并有利于位同步时钟提取。扰码方法很多，读者可以自定义，也可以采用标准的4bit/5bit编码。标准的4bit/5bit编码见表2-16。

表2-16　4b/5b编码对照表

数据序列	编码序列	数据序列	编码序列
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

2. 设计内容

(a) 进行编解码器需求分析，定义信号及属性。

(b) 设计编码方案（自定义情况）。

(c) 设计和分析编码器与解码器的电路原理，给出电路原理图。

(d) 设计测试方法和测试数据，安装电路，并调试。

3. 设计要求

(a) 输入数据宽度为N*4bit（N=1，2）。

(b) 编码器数据并行输入、串行输出，解码器数据串行输入、并行输出。

(c) 为方便调试，将中间串行数据接至LED。

(d) 相关信号之间满足工作时序要求。

(e) 按要求完成课程设计报告。

4. 设计提示

首先，待扰码数据在锁存控制信号的作用下存入编码器内部数据锁存器，然后，由内部逻辑电路同步编码后，在内部时序控制下加载到内部移位寄存器，最后，移位寄存器在时钟信号CLK控制下逐位输出（并串转换），完成编码过程。电路框图见图2-41。每当移位寄存器完成全部比特位输出时就会产生一个状态信号，用于指示可以输入新数据。解码器完成编码逆过程，读者可以自己设计。

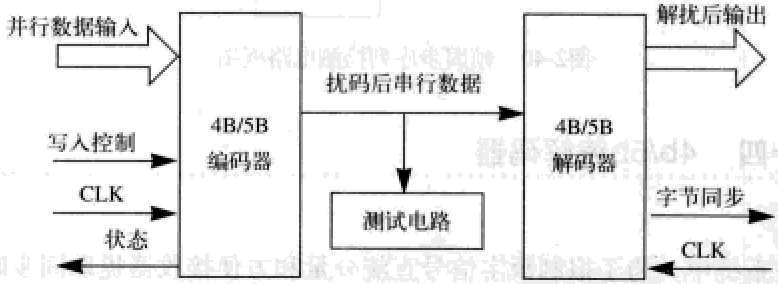


图2-41 4b/5b编解码器电路框图

课程设计二十五 数字秒表

1. 设计说明

在体育比赛和时间精确测量等场合通常要求计时精度到1%秒（即10ms）甚至更高的计时装置。数字秒表是一种精确的计时仪表，可以担当此任。本课程设计的任务是设计一个以数字方式显示的计时器，即数字秒表。

2. 设计内容

(a) 进行数字秒表需求分析，定义信号及属性。

(b) 设计和分析电路原理、计算参数，画出电路原理图。

(c) 安装电路，并调试。

3. 设计要求

(a) 量程99.99s，计时精度1%秒，计时结果动态显示，十进制格式。

(b) 设置启动、清除信号，清除信号使电路输出结果，使电路复位到初始状态。

(c) 设置暂停、停止信号，暂停、停止时均保持当前结果，直到清除信号有效时止。

(d) 按要求完成课程设计报告。

4. 设计提示

数字秒表由4个部分组成：精确的时钟源、十进制计数器、译码器、七段数码管或液晶显示电路。电路原理框图如图2-42所示。

时钟源产生符合精度要求的基准时钟，本设计中取10ms即可。十进制计数器需要4个，分别对应4个十进制位，输出为BCD码。若采用七段数码管则译码器完成BCD到七段数码管的译码，由4位显示电路动态显示结果。

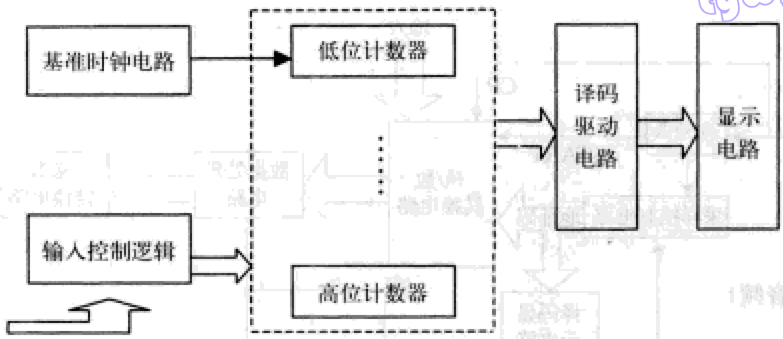


图2-42 数字秒表电路原理框图

课程设计二十六 数据采集与监测系统

1. 设计说明

随着计算机应用的普及，将模拟信号转化成数字信号以及将数字信号转化成模拟信号就成为应用的重要环节。本课程设计以多路数据的采集及监测为例，介绍了模/数转换、数/模转换、数据处理、数字量及模拟量显示等环节的基本原理及实施方法。多路信号采集与监测系统能对输入的多路模拟信号进行定时循环采样，将它们变为数字量。同时，对采集到的数字信号进行数据处理，判断该输入信号是否正常。若正常，将其转换成相应的模拟量输出；若不正常，输出规定的模拟量，同时进行声光报警，并将采样停留在该路信号上等待处理。因此，多路信号采集与监测系统应具有采样控制电路、模/数转换电路、译码显示电路、数据处理电路、数/模转换电路、声光报警电路及脉冲信号产生电路。本课程设计的任务是要设计一个数据采集与监测系统。

2. 设计内容

(a) 进行数据采集与监测系统需求分析，定义信号。

(b) 设计和分析电路原理，画出电路原理图。

(c) 安装电路，并调试。

3. 设计要求

(a) 设计一路模拟信号采样电路，要求采样间隔为5~10s。

(b) 设计采样信号上下限比较电路。上下限可以自行设定。要求当采样数据在上下限之间时，经过数/模转换输出采样信号的模拟量；当采样数据在超限时，使数/模转换为零。

(c) 设计超限报警电路。要求输入超上限时，红灯亮，喇叭响；输入超下限时，黄灯亮，喇叭响。

(d) 设计4路模拟信号循环采样电路。要求输入信号正常时，采样在该路信号上停留5~10s后再进行下一路信号的采样；当输入模拟信号超限时，采样一直停留在该路信号上。

(e) 设计采样通道号显示电路，要求用七段数码管显示当前采样通道号。

4. 设计提示

其结构框图如图2-43所示。

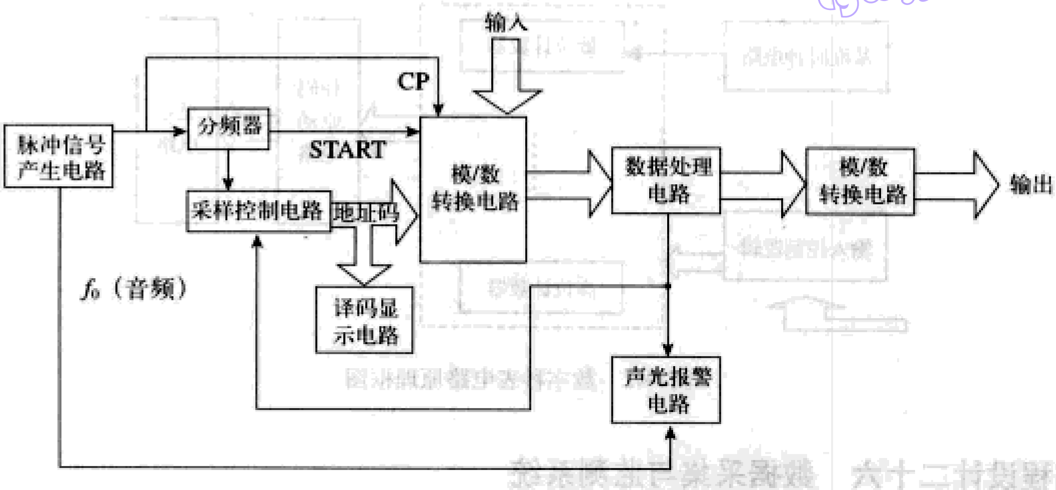


图2-43 数据采集与监测系统电路框图

第3章　可编程器件应用实验与课程设计

3.1　EDA工具的使用

3.1.1　QuartusII软件的使用

QuartusII是Altera公司推出的CPLD/FPGA开发工具，它提供了完全集成且与电路结构无关的开发包环境，具有数字逻辑设计的全部特性，包括：可利用原理图、结构框图、VerilogHDL、AHDL和VHDL完成电路描述，并将其保存为设计实体文件；编辑芯片（电路）平面布局连线；用LogicLock增量设计方法，用户可建立并优化系统，然后添加对原始系统的性能影响较小或无影响的后续模块；可使用SignalTap II逻辑分析工具进行嵌入式的逻辑分析；支持软件源文件的添加和创建，并将它们链接起来生成编程文件；使用组合编译方式可一次完成整体设计流程；自动定位编译错误；可读入标准的EDIF网表文件、VHDL网表文件和Verilog网表文件；能生成第三方EDA软件使用的VHDL网表文件和Verilog网表文件。还包括功能强大的逻辑综合工具，完备的电路功能仿真与时序逻辑仿真工具，定时/时序分析与关键路径延时分析，高效的期间编程与验证工具。

这里将以4位二进制加法计数器为例来介绍QuartusII的使用方法，实验中用到的实验箱为GW48-PK2。用QuartusII设计一个4位二进制加法计数器的具体步骤如下。

一、创建工程和编辑设计文件

首先应该建立好工作库目录。此例中设立文件夹为E：\vtest\count4作为工作库，以便将设计过程中的相关文件存储在此。任何设计都是一项工程（Project），都必须建立一个放置与此工程相关的文件的文件夹，此文件夹将被EDA软件默认为工作库（Work Library）。注意，文件夹名不能用中文。

在建立文件夹后就可以创建工程和编辑设计文件了，具体步骤如下。

（1）利用“New Preject Wizard”创建此设计工程。选择菜单“File”–>“New Preject Wizard”，点击Next，即可弹出工程设置对话框（图3-1）。点击此框最上一栏右侧的按钮“...”，设置工程路径，找到文件夹E:\vtest\count4，填写工程名和顶层文件名称后，点击Next按钮进行下一步。

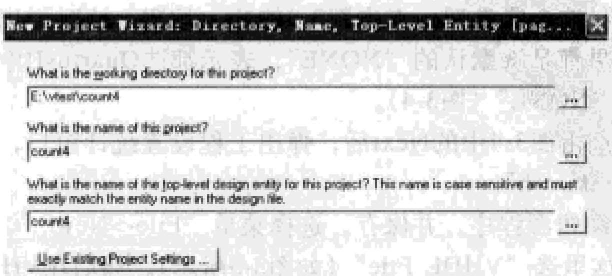


图3-1　工程路径设置对话框

（2）添加设计源程序。如图3-2所示，如果已有源程序，可以在此加入到工程中，如果没有就点击Next进行下一步。

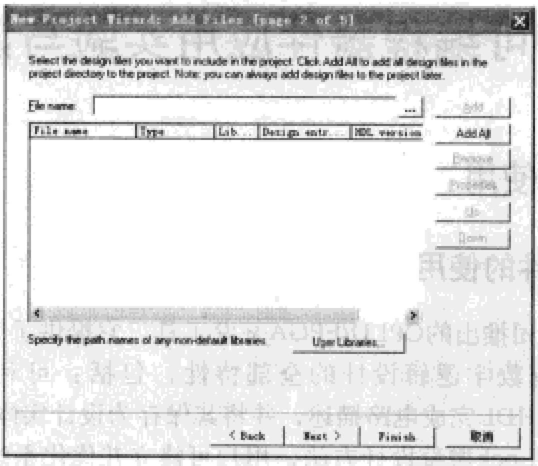


图3-2　添加设计文件对话框

（3）选择目标芯片。首先在“Family”栏选芯片系列，在此选“ACEX1K”系列，选择此系列的具体芯片：EP1K30TC144-3（图3-3）。

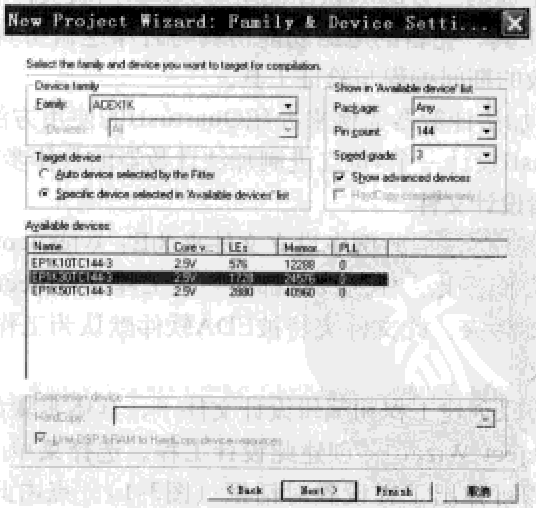


图3-3　选定目标器件对话框

（4）选择仿真器和综合器类型。点击图3-3的Next按钮，这时弹出的窗口用于选择仿真器和综合器类型。如果都是选默认的“NONE”，表示都选QuartusII中自带的仿真器和综合器。在此都选默认项“NONE”（图3-4）。

（5）结束设置。点击图3-4中的Next后，弹出工程设置统计窗口，列出此项工程的相关设置情况。最后点击“Finish”。

（6）在Quartus中编辑源程序，并保存。选择菜单“File->New...”，在New窗口中选择编译文件的语言类型，这里选“VHDL File”（如图3-5所示）。然后在VHDL文本编译窗口中键入如图3-6所示的VHDL程序，将其存盘为count4.vhd。

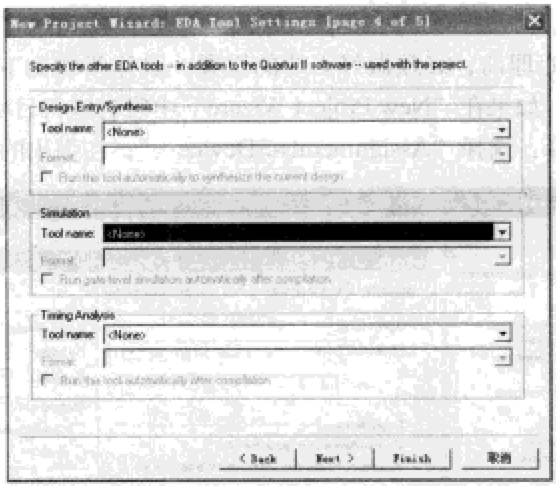


图3-4 设置综合器、仿真器、时序分析器类型对话框

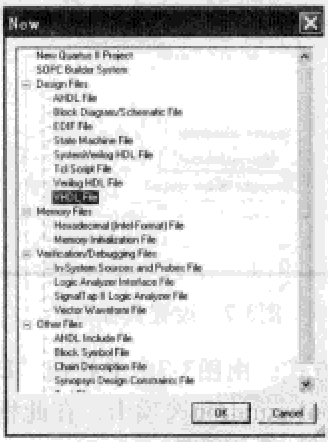


图3-5 选择编辑文件的语言类型

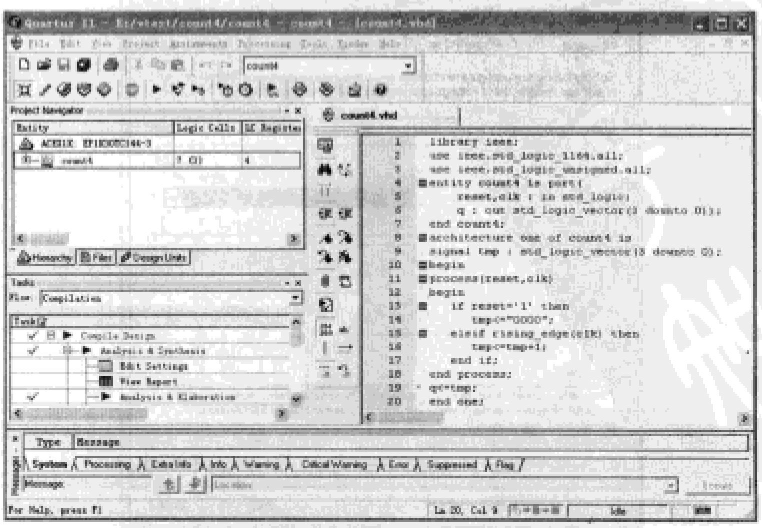


图3-6 编辑输入设计文件

二、编译前设置

在对工程进行编译处理前，必须做好必要的设置。具体步骤如下。

（1）选择目标芯片。如果在“New Project Wizard”中没有进行目标芯片的设置，也可以这样来进行目标芯片的选择：选择“Assignmemts->Device…”，在弹出的对话框中选择Device。

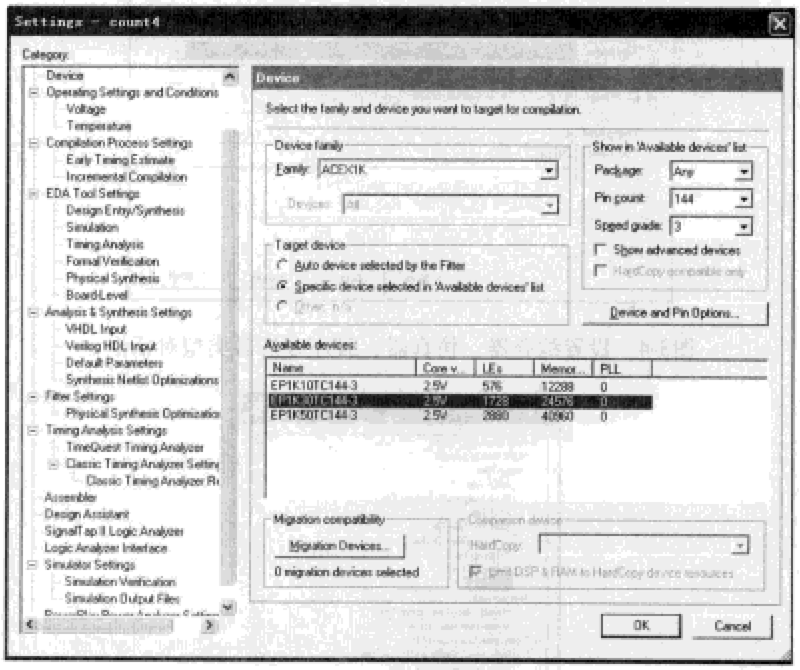


图3-7　设置目标器件

（2）选择目标器件编程配置方式。由图3-7中的按钮“Device and Pin Options”进入选择窗口，如图3-8所示。首先点击Configuration选项卡，在此框的下方有相应的说明，在此可选

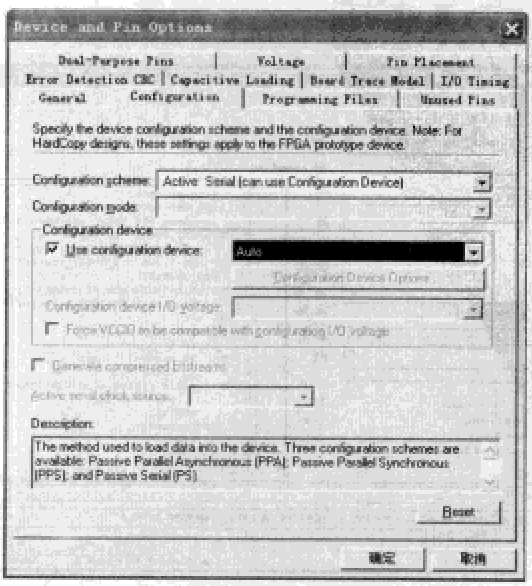


图3-8　选择配置器件和配置方式

Configuration方式为Active Serial，这是对专用配置器件进行配置的编程方式，而PC机对此FPGA的直接配置方式都是JTAG方式的。在“Configuration device”项，选择配置器，如果电路中没有配置器件，可选择Auto（根据实验系统上目标器件配置的EPCS芯片决定，如图3-8所示）。

（3）对未使用的引脚进行设置。在图3-9的“Unused Pins”项中，对未使用的引脚进行设置。具体设置可根据电路板的情况而定。

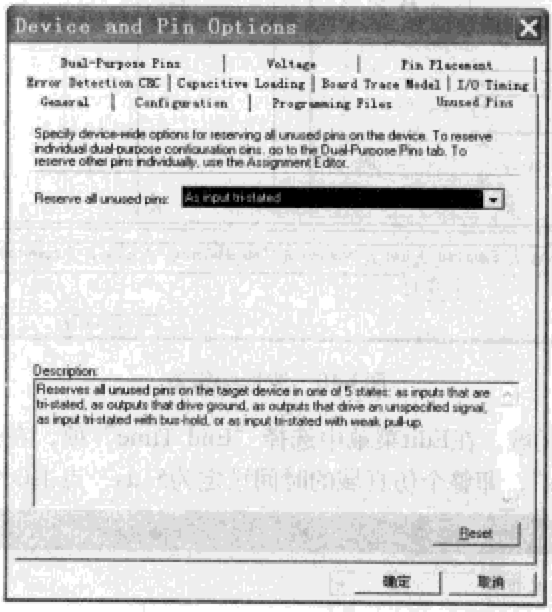


图3-9 设置未使用的引脚

三、编译

选择Processing菜单的“Start Compilation”项，启动全程编译。注意这里所谓的编译（Compilation）包括QuartusII对设计输入的多项处理操作，其中包括排错、数据网表文件提取、逻辑综合、适配、装配文件（仿真文件与编程配置文件）生成，以及基于目标器件的工程时序分析等。如果工程中的文件有错误，在下方的Processing处理栏中会显示出来。如果Processing栏显示出语句格式错误，可双击此条，即弹出VHDL文件，在闪动的光标处（或附近）可发现文件中的错误。再次进行编译直至排除所有错误。

四、仿真

仿真就是对设计项目进行全面彻底的测试，以确保设计项目的功能和时序特性，以及最后的硬件器件的功能与原设计相吻合。对工程的编译通过后，必须对其功能和时序性质进行仿真测试，以了解设计结果是否满足原设计要求。仿真操作前必须利用QuartusII的波形编辑器建立一个矢量波形文件以作仿真激励，步骤如下。

（1）打开波形编辑器。选择菜单File中的New项，在New窗口中选“Vector Waveform File”，点击OK，即出现空白的波形编辑器（图3-10）。

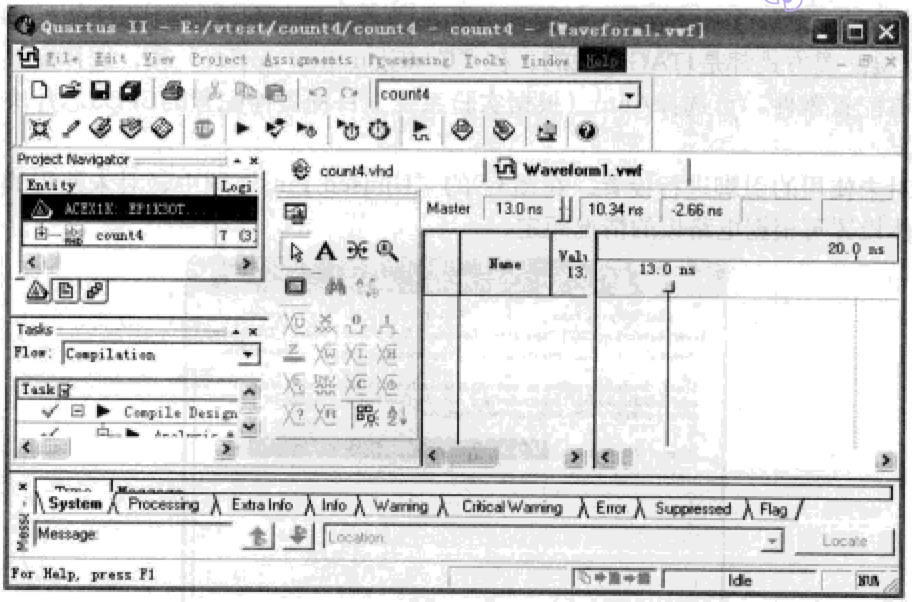


图3-10　波形编辑器

（2）设置仿真时间区域。在Edit菜单中选择“End Time”项，在弹出的窗口中的“Time”窗中键入51，单位选“us”，即整个仿真域的时间设定为51μs，点击OK，结束设置，见图3-11。

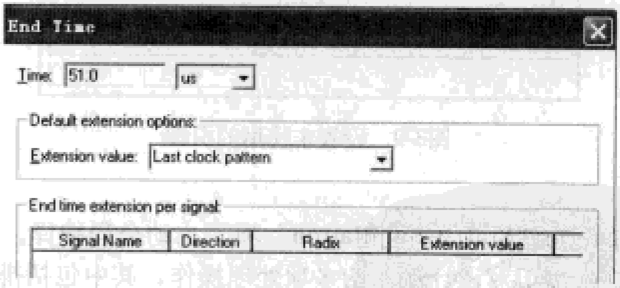


图3-11　设置仿真时间

（3）输入信号节点。将计数器的端口信号节点选入此波形编辑器中。方法是首先选View菜单中的“Utility Windows”项的“Node Finder”选项（也可使用快捷键Alt-L快速调出Node Finder窗口）。其对话框如图3-12所示，在Filter框中选Pins：all，然后点击“List”。于

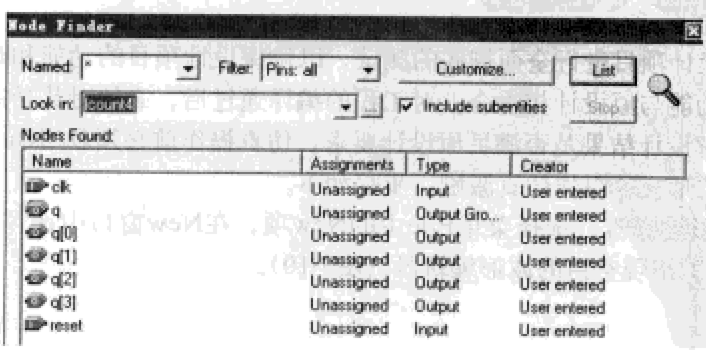


图3-12　Node Finder对话框

是在下方的“Nodes Found”窗中出现了设计中的count4工程的所有端口引脚名。用鼠标将端口节点CLK、RESET和输出总线信号Q都拖到波形编辑窗，点击放大缩小按钮后，在波形编辑区域右击鼠标，使仿真坐标处于适当位置（图3-13）。

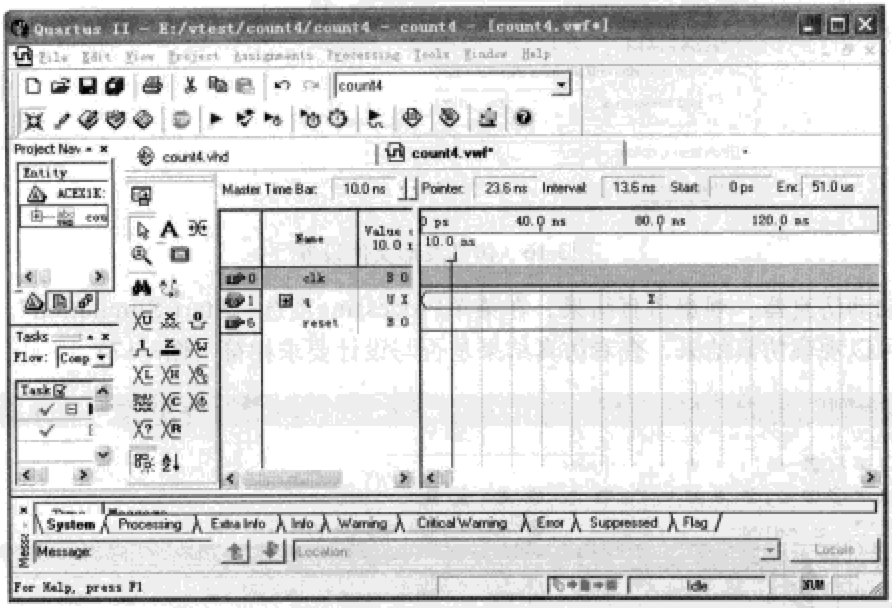


图3-13　设置好输入输出端口的波形编辑器

（4）编辑输入波形。点击时钟名CLK，使之变成蓝色，再点击左列的时钟设置键，在Clock窗中设置CLK的周期为20ns。图3-14所示的Clock窗中的“Duty cycle”是占空比，可选50，即50%占空比。选择一段范围的RESET信号，再点击左列的高电平设置键，使RESET在某段时间为高电平，实现电路的复位。

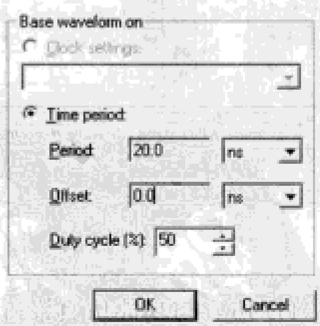


图3-14　选择时钟周期和占空比

（5）总线数据格式设置。点击输出总线信号Q，使之变蓝色，在Q上右击，在弹出菜单中选择Properties菜单，在“Node Properties”对话框中的Radix中选择Unsigned Decimal（图3-15）。

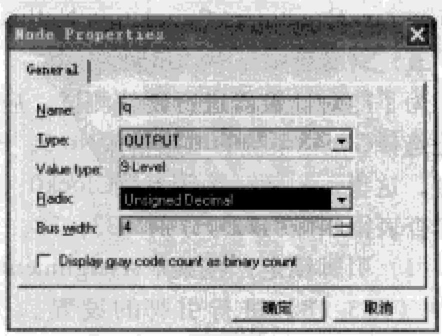


图3-15　设置仿真信号数据格式

（6）存盘波形文件。选择File中的“Save as”，将以名为count4.vwf（默认名）的波形文件存入文件夹E:\vtest\count4中。注意，如果仿真文件名与工程默认名称不同，例如取名为cnt4.vwf，则要在工程中进行相应的设置才能正常仿真。设置方法是，在菜单Assignments中选Settings...，然后在弹出的对话框的Simulator Settings的Simulation Input中进行设置（图3-16）。

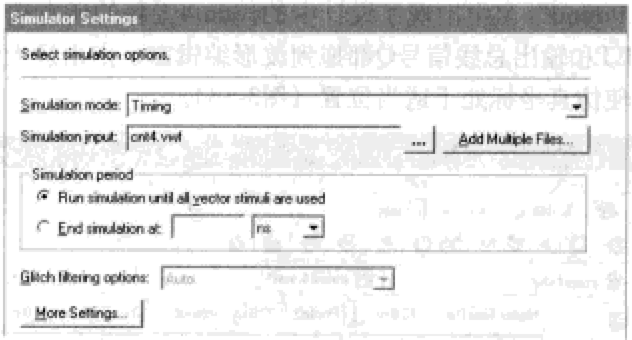


图3-16　仿真波形文件设置

（7）启动仿真器、观察仿真结果。在菜单Processing项选“Start Simulation”，仿真结果出来后，可以观察仿真结果，查看仿真结果是否与设计要求相符（图3-17）。

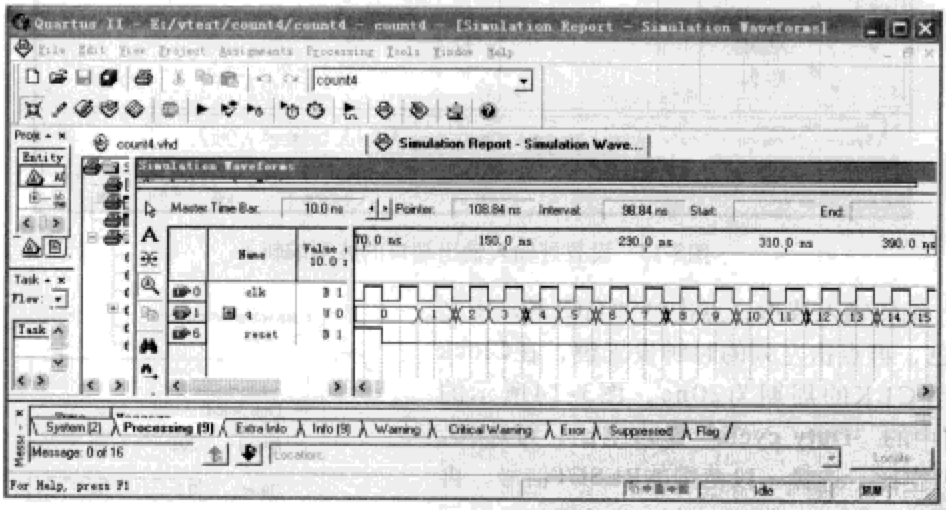


图3-17　4位二进制加法计数器仿真结果

（8）时序仿真与功能仿真设置。QuartusII既可以进行时序仿真，又可以进行功能仿真，默认的仿真类型是时序仿真。如果要进行功能仿真也要进行相应的设置。具体方法是，将Simulator Settings（图3-16）中的Simulation Mode设置成Functional，在菜单Processing中选择Generate Functional Simulation Nellist，然后，就可以进行功能仿真了。

五、引脚锁定、下载和硬件测试

为了能对计数器进行硬件测试，应将计数器的输入输出信号锁定在芯片确定的引脚上。在此选择GW48系统的电路模式No.5。EP1K30适配板的引脚情况，通过查相关引脚对照表来确定。这里，主频时钟CLK接Clock0（第126脚）；4位二进制的计数输出Q由数码管1显示，通过查表得相应引脚号分别为33、32、31、30；RESET信号由键1控制，查得引脚号为8。

（1）引脚锁定。在菜单Assignments中，选Pins项或Assignments Editor项，在弹出的窗口中（图3-18）进行引脚的设置。引脚锁定后，必须再编译一次（Processing Start Compilation），将引脚锁定信息编译进下载文件中。

（2）选择编程器。为了将编译产生的下载文件配置进FPGA中进行测试，首先要将系统

连接好，上电，然后在菜单Tool中，选择Programmer，于是弹出如图3-19所示的编程窗口。在此编程窗口中，点击“Hardware Setup”，选择编程器类型。这里选择“ByteBlasterMV”。

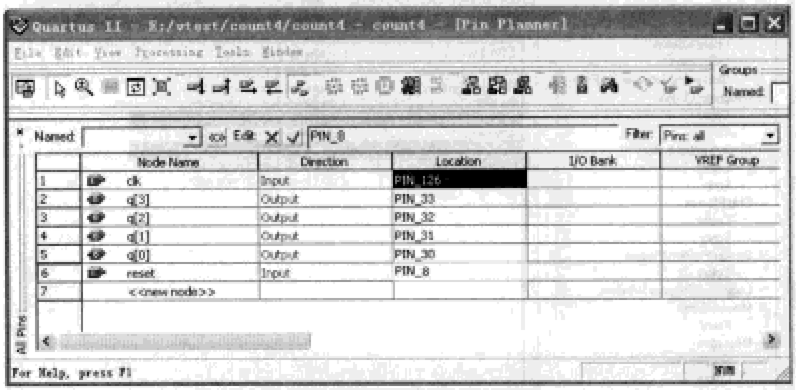


图3-18 引脚锁定图

（3）选择编程模式和配置。在图3-19所示的编程窗口中，Mode栏中有三种编程模式可以选择，JTAG、Passive Serial和Active Serial。选JTAG，最后点击下载符。当“Progress”显示出100%，而且在底部的处理栏中出现“Configuration Succeeded”时，表示编程成功。为了能很好地观察到实验结果，可以将实验箱的clock0的跳线帽设置1Hz。

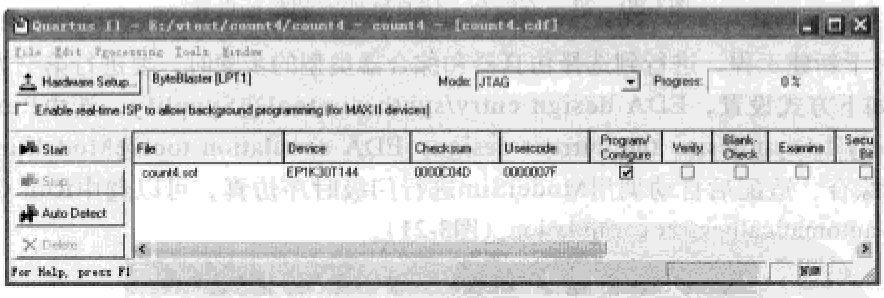


图3-19 编程下载对话框

3.1.2 QuartusII中集成第三方软件的使用

从QuartusII 7.1开始支持Node-Locked License的第三方综合器直接从Quartus中调用，因此，我们可以更加方便地在QuartusII中使用第三方软件。下面就举例说明如何从Quartus中直接调用第三方的综合器和仿真器进行综合与仿真。综合器以Synplify为例，仿真器以Modelsim为例。例子中的VHDL程序仍采用4位二进制加法计数器为例。Quartus采用7.1，安装在D:\altera；Synplify版本是Synplify 8.6.2，安装在D:\Synplicity；Modelsim版本是ModelSim SE 6.2e，安装在D:\Modeltech_6.2e。具体步骤如下。

（1）第三方软件路径的设置。要想在QuartusII中直接调用第三方的工具软件，就必须让QuarusII知道第三方软件的路径位置，设置路径的方法如图3-20所示，在菜单Tools->Options->General->EDA Tools Options中设置Synplify和ModelSim的安装路径。注意，同时要选中Enable NativeLink for Synplify/Synplify Pro with a node-locked license。这一点非常重要，如果不选的话，就无法从Quartus中直接调用Synplify进行综合。

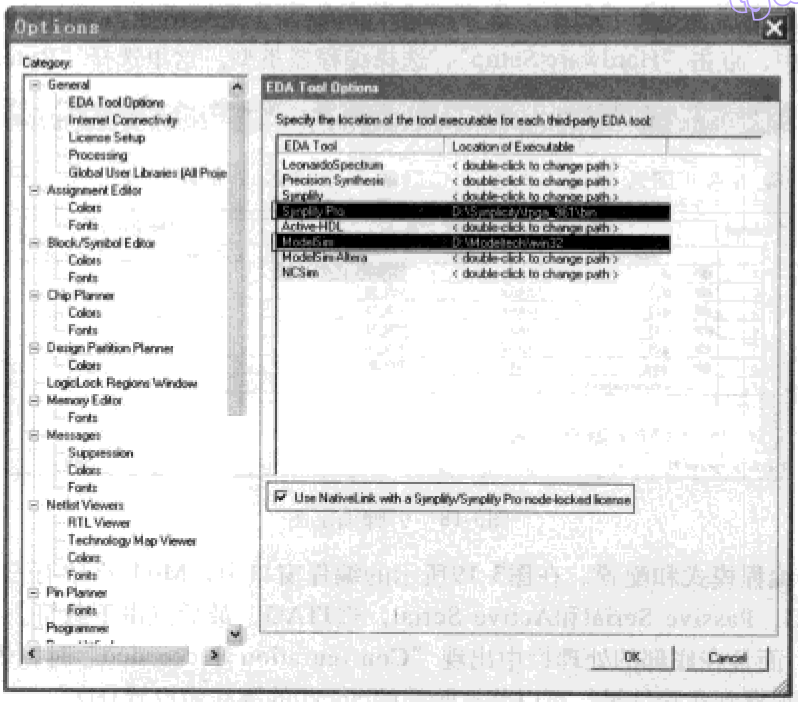


图3-20　第三方综合、仿真软件安装路径设置

（2）对于新建工程，进行到选择仿真器和综合器类型的步骤时，要进行第三方工具的设置。按照如下方式设置，EDA design entry/synthesis tool选Synplify，选中Run this tool automatically to synthesize the current design；EDA simulation tool选ModelSim，如果让Quartus在综合、适配后自动调用ModelSim进行门级时序仿真，可以选中Run Gate Level Simulation automatically after compilation（图3-21）。

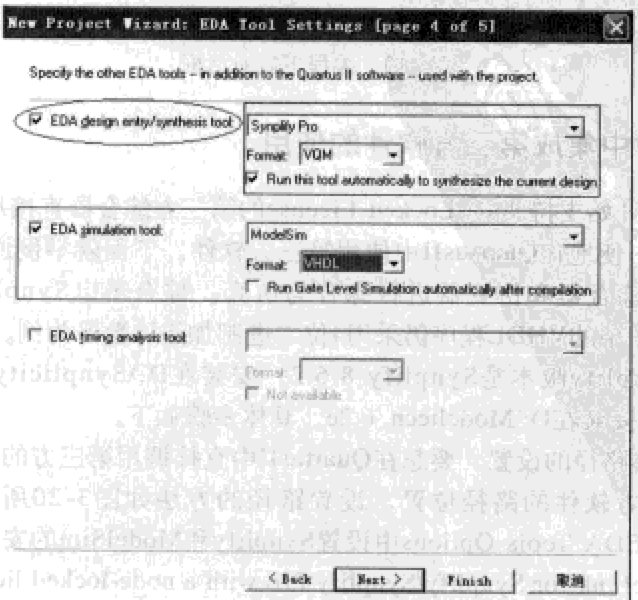


图3-21　设置第三方综合器、仿真器

(3) 对于工程的文件要进行ModelSim仿真，我们还需要编写一个TestBench程序，假设对于上面的工程，我们编写的Testbench如下，并把它存为count_tb_vhd.vhd。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity count_tb_vhd is
  generic  (clk_period : Time := 20 ns) ;
end count_tb_vhd;
architecture behave of count_tb_vhd is
component count4
port
       (reset,clk:in std_logic;
       q:out std_logic_vector (3 downto 0)) ;
end component;
signal clk:std_logic:='0';
signal reset:std_logic:='0';
signal q:std_logic_vector (3 downto 0) ;
--constant clk_period:time:=20ns;
begin
u1:count4 port map
(clk=>clk,
reset=>reset,
q=>q) ;
clk_gen: process
begin
clk<='1';
wait for clk_period/2;
clk<='0';
wait for clk_period/2;
end process;
tb: process
begin
wait for 20ns;
reset<='1';
wait for 20ns;
reset<='0';
wait for 200ns;
wait;
end process;
process
begin
    wait for clk_period * 5000;
    wait for clk_period / 2;
    assert false report
    "ENDOFSIMULATION"
```

```
    severity failure;
end process;
end behave;
```

（4）在Quartus中对Testbench进行相应设置。选择菜单Assignments->Settings->EDA Tool Settings->Simulation选中Compile test bench（图3-22），点击Test Benches...按钮进入下一界面。

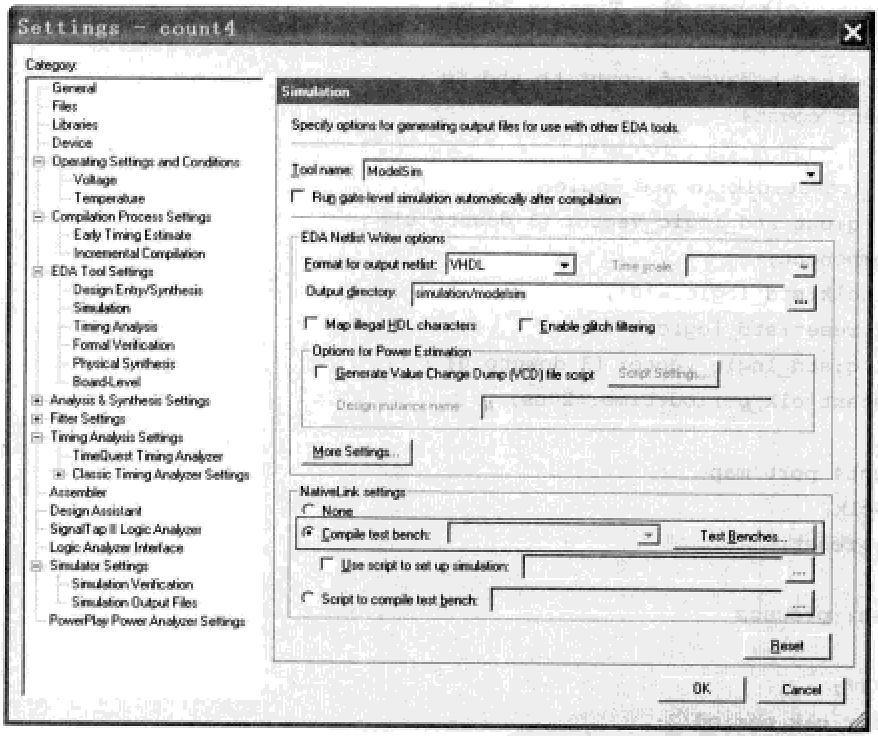


图3-22　Testbench文件设置

（5）按下New...进行下一步设置（图3-23）。

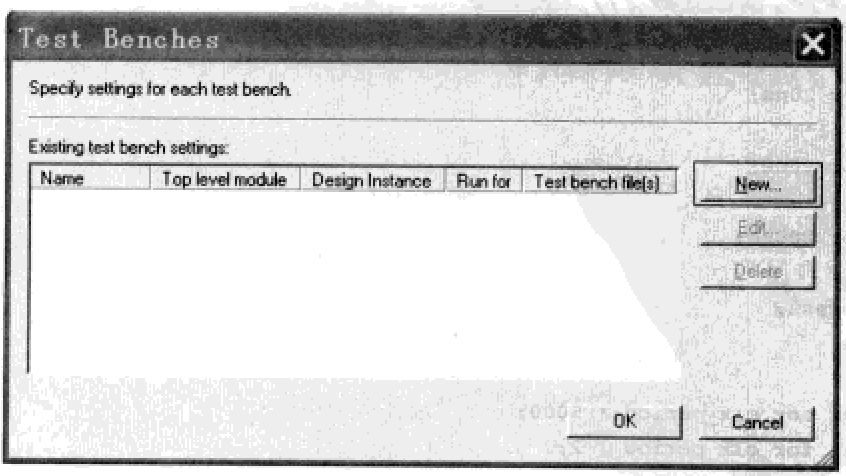


图3-23　新建Testbench

(6) 分别填入测试名称、顶层模块名和例化元件的名称u1，点击File name后的□按钮，设置testbench文件位置（图3-24），选择count_tb_vhd.vhd（图3-25），并将测试文件加入列表中（图3-26）。图3-27是设置完成后的示意图。至此，我们已经完成了第三方软件的设置，下面就可以用第三方工具进行综合和仿真等工作了。

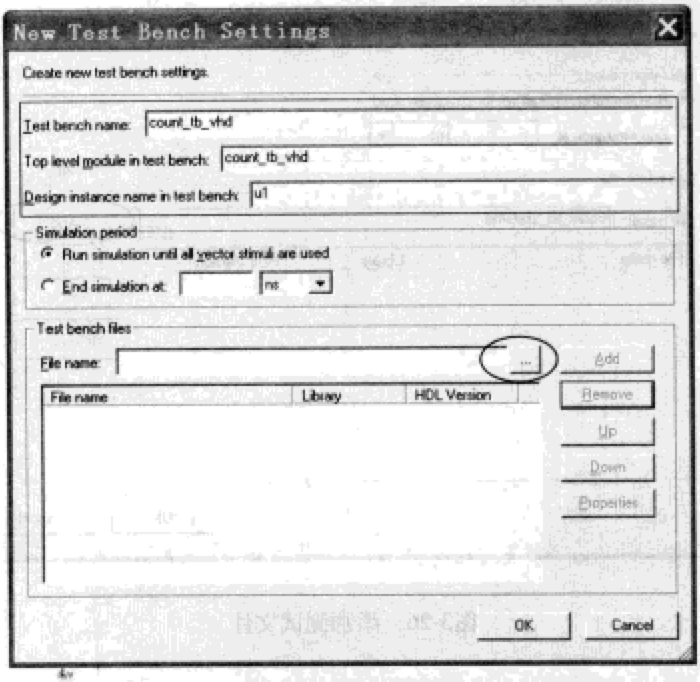


图3-24 Testbench详细设置

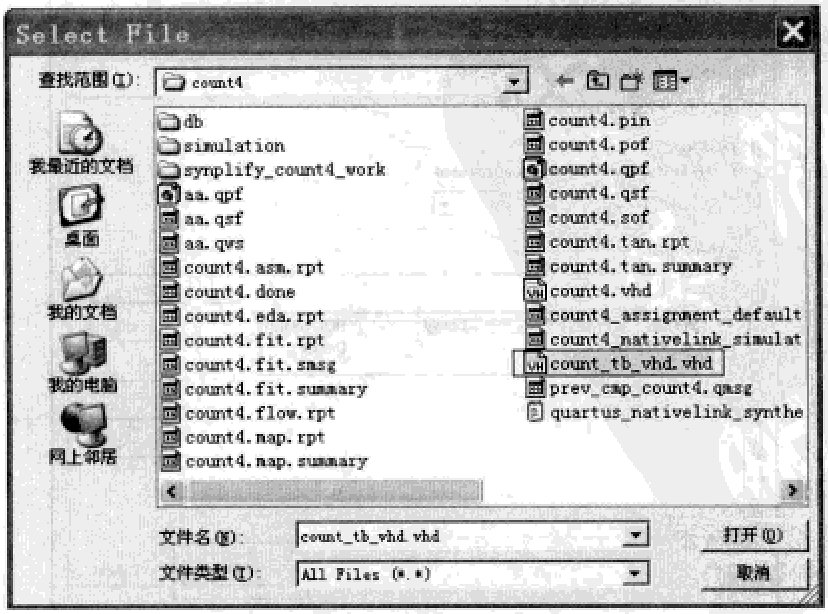


图3-25 选择Testbench文件位置

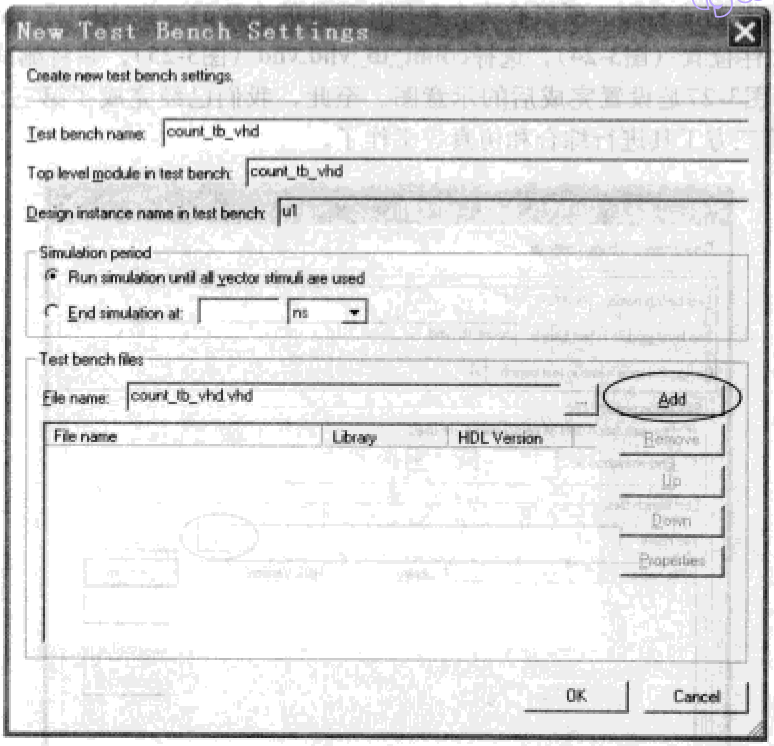


图3-26　添加测试文件

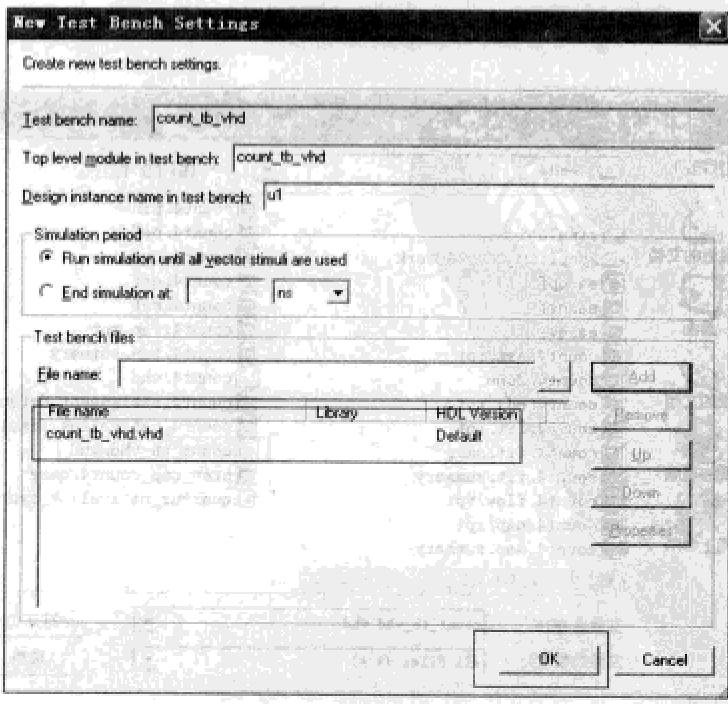


图3-27　设置完成后的图例

（7）利用第三方软件进行综合与仿真。执行菜单Processing->Start Compilation，Quartus就会自动调用Synplify进行综合，然后再自动完成适配、时序分析和写网表等工作。如果在第三方工具的设置中（图3-21）选中了Run Gate Level Simulation automatically after compilation，它还会自动调用Modelsim完成门级时序仿真，并自动显示出仿真波形。如果没有选中Run Gate Level Simulation automatically after compilation，则可以在Quartus中选择菜单Tools->Run EDA Gate Level Simulation, Quartus会调用Modelsim进行门级时序仿真，并自动显示出仿真波形（图3-28）。

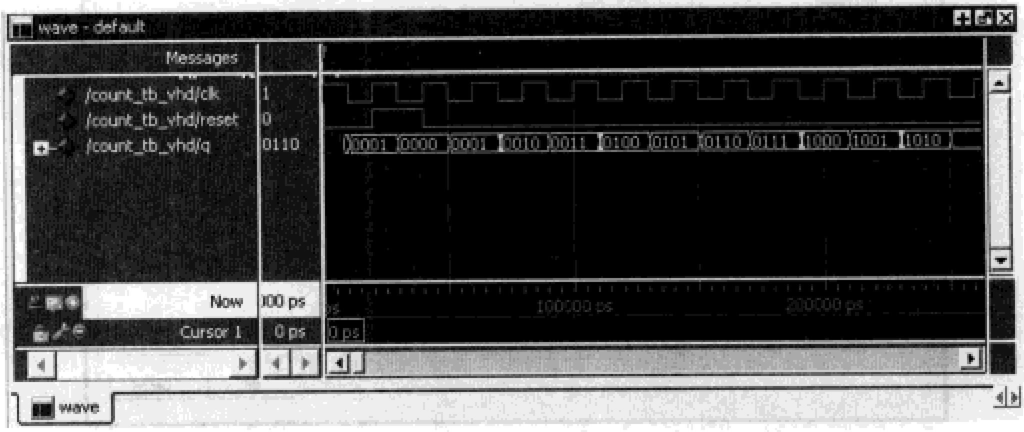


图3-28 门级时序仿真结果（综合器可以是第三方软件）

如果综合器没有选择第三方的工具，而是直接用Quartus自己的综合器。还可以让Quartus调用Modelsim执行RTL级功能仿真。在Quartus中选择菜单Tools->Run EDA RTL Simulation，则Quartus会调用Modelsim进行RTL级功能仿真，并自动显示出仿真波形（图3-29）。

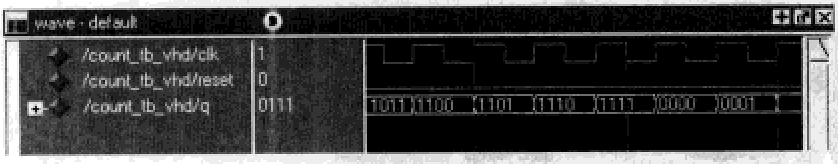


图3-29 RTL级功能仿真结果（采用Quartus自己的综合器）

（8）如果在创建工程时没有对第三方软件进行设置，也可以在创建工程后再进行设置。

设置综合器可以选择菜单Assignments->Settings->EDA Tool Settings->Design Entry/Synthesis。Tool name选择Synplify Pro，并且选中Run this tool automatically to synthesize the current design（图3-30）。

设置仿真器可以选择菜单Assignments->Settings->EDA Tool Settings->Simulation。Tool name选择Modelsim，并且选中Run this tool automatically to synthesize the current design（图3-31）。

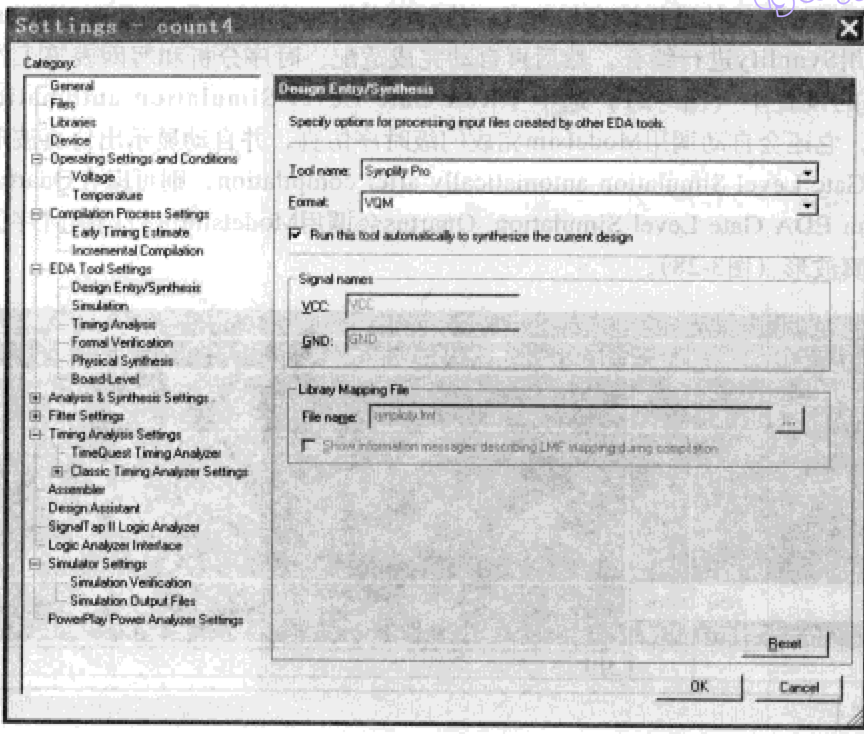


图3-30　第三方综合工具的设置

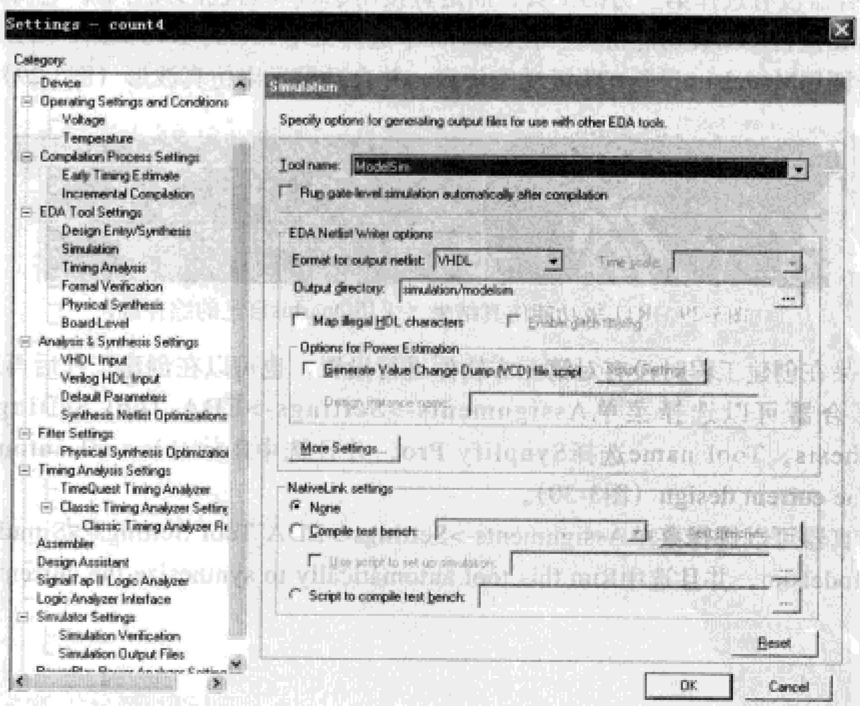


图3-31　第三方仿真工具的设置

3.2 可编程器件基本实验

实验一 1位全加器VHDL文本输入设计

一、实验目的

（1）学习使用VHDL语言设计简单组合逻辑电路。

（2）学习使用VHDL语言中的结构描述方法设计逻辑电路。

（3）学习使用QuartusII软件。

（4）学习FPGA器件的编程和配置。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

利用QuartusII设计一位全加器（参考程序如下），仿真测试，给出仿真波形，进行引脚锁定,并在实验箱上进行硬件验证。

实验具体步骤如下。

（1）用原理图方式或文本方式输入设计文件并存盘。

（2）选择目标器件并编译、综合。

（3）进行时序仿真。

（4）锁定引脚，并编译、综合。

（5）在实验箱上选择模式5或6。

（6）将数据下载到实验箱。

（7）在实验箱上验证全加器的功能。

```
--程序1：或门逻辑描述
 LIBRARY  IEEE ;
 USE IEEE.STD_LOGIC_1164.ALL;
 ENTITY or2a IS
   PORT  (a, b :IN STD_LOGIC;
               c : OUT STD_LOGIC) ;
 END ENTITY or2a;
 ARCHITECTURE one OF or2a IS
   BEGIN
   c <= a OR b ;
 END ARCHITECTURE ful;

--程序2 ：半加器描述
LIBRARY  IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY h_adder IS
  PORT  (a, b : IN STD_LOGIC;
       co, so : OUT STD_LOGIC) ;
END ENTITY h_adder;
```

```
ARCHITECTURE fh1 OF h_adder is
BEGIN
  so <= NOT (a XOR  (NOT b)) ;
  co <= a AND b ;
END ARCHITECTURE fh1;

--程序3 ： 1位二进制全加器顶层设计描述
 LIBRARY  IEEE;
 USE IEEE.STD_LOGIC_1164.ALL;
 ENTITY f_adder IS
   PORT  (ain, bin, cin  : IN STD_LOGIC;
          cout, sum    : OUT STD_LOGIC) ;
 END ENTITY f_adder;
 ARCHITECTURE fd1 OF f_adder IS
   COMPONENT h_adder
     PORT ( a, b :  IN STD_LOGIC;
         co, so :  OUT STD_LOGIC) ;
   END COMPONENT ;
   COMPONENT or2a
      PORT  (a, b : IN STD_LOGIC;
             co, so c : OUT STD_LOGIC) ;
   END COMPONENT;
SIGNAL d, e, f   :  STD_LOGIC;
  BEGIN
   u1  : h_adder PORT MAP (a=>ain, b=>bin,
         co=>d, so=>e) ;
   u2  : h_adder PORT MAP (a=>e,  b=>cin,
         co=>f, so=>sum) ;
   u3  : or2a PORT MAP (a=>d,  b=>f, c=>cout) ;
 END ARCHITECTURE fd1 ;
```

注意：此3个程序必须分别进行编译、设置成工程和仿真，最后处理顶层文件程序3。

四、实验报告要求

(1) 写出用VHDL语言设计一位全加器的设计方案。

(2) 用QuartusII作出一位全加器的仿真波形，并与实验结果进行对比。

(3) 写出VHDL结构描述方法设计逻辑电路的心得。

实验二　7人投票表决器VHDL设计

一、实验目的

(1) 学习使用VHDL语言设计组合逻辑电路。

(2) 学习使用VHDL语言中的结构描述方式与行为方式设计逻辑电路。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

利用QuartusII设计一个7人投票表决器，具体要求为：参加表决者7人，同意为1，不同意为0，同意者过半则表决通过，指示灯亮。在QuartusII环境下，输入设计方案，并进行编译、综合，给出仿真波形，进行引脚锁定，然后下载到实验箱，并在实验箱上进行硬件验证。

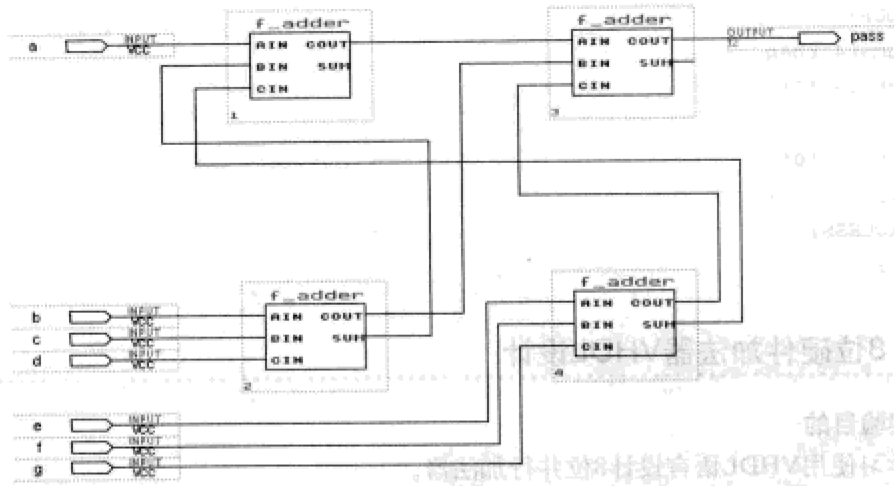


图3-32 7人投票表决器原理图

注1：可以采用1位全加器构成，其原理图如图3-32所示，也可以采用VHDL语言的行为描述方式进行设计。

注2：建议选“实验电路模式5”，用键1～7作为7个表决输入，表决结果接发光管D8，引脚锁定后进行编译、下载和硬件测试实验。

四、实验报告要求

(1) 写出用VHDL语言设计7人投票表决器的设计方案。
(2) 用QuartusII作出7人投票表决器的仿真波形，并与实验结果进行对比。
(3) 将实验过程和实验结果写进实验报告。
(4) 如果要求同时显示投赞成票的人数，应如何修改设计。
(5) 写出VHDL结构描述方式与行为描述方式设计逻辑电路的心得。

```
LIBRARY ieee;
USE ieee.Std_logic_1164.ALL;
USE ieee.Std_logic_unsigned.ALL;
ENTITY vote7 IS
PORT (
datain : IN Std_logic_vector (6 DOWNTO 0) ;
termcnt : OUT Std_logic) ;
END vote7;
ARCHITECTURE v1 OF vote7 IS
BEGIN
main_proc : PROCESS (datain)
variable tsum : integer range 0 to 7;
BEGIN
```

```
tsum := 0;
FOR i IN datain' Range LOOP
IF datain (i) = '1' THEN
tsum := tsum+1;
END IF;
END LOOP;
if tsum>=4 then
termcnt <= '1' ;
else
termcnt <= '0' ;
end if;
END PROCESS;
END v1;
```

实验三　8位硬件加法器VHDL设计

一、实验目的

（1）学习使用VHDL语言设计8位并行加法器。

（2）学习各种加法器的设计方法，并比较几种加法器设计方法的特点。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

利用QuartusII设计一个8位加法器，具体要求为，设计一个8位加法器，并比较采用4位加法器和串行级联加法器在资源占用及速度方面的情况。

实验具体步骤如下。

（1）在QuartusII环境下，对两种方案进行编译、综合、适配、下载，验证结果。

（2）进行时序仿真。

（3）查看两种方案的资源占用情况。

设计提示：可选择模式1，加数用键2/1输入，被加数用键4/3输入，数码6/5显示和，键8输入进位输入，D8显示进位输出。

四、实验报告要求

（1）写出用VHDL语言设计8位加法器的设计方案。

（2）用QuartusII作出8位加法器的仿真波形，并与实验结果进行对比。

（3）比较不同方法设计加法器的资源占用情况。

（4）将实验过程和实验结果写进实验报告。

实验四　设计一个含异步清0和同步时钟使能的4位加法计数器

一、实验目的

（1）学习使用VHDL语言设计时序电路。

（2）了解掌握异步与同步电路的设计方法。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

计数器是常用的时序电路，图3-33是一个含计数使能、异步复位和计数值并行预置功能的4位加法计数器，下面的程序是其VHDL描述。由图3-33所示，中间是4位寄存器；RST是异步清除信号，高电平有效；CLK是时钟信号；D[3..0]是4位数据输入端。当ENA为“1”时，多路选择器将累加器的输出值加载于寄存器的数据端；当ENA为“0”时，寄存器保持原值。本次实验的内容就是利用QuartusII设计一个加法计数器，具体要求为，设计一个含异步清0和同步时钟使能的4位加法计数器。

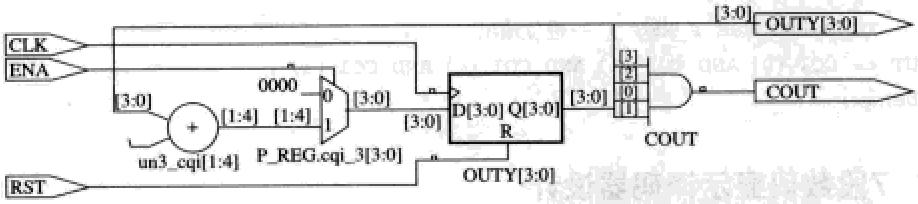


图3-33　含计数使能、异步复位和计数值并行预置功能的4位加法计数器

实验具体步骤如下。

(1) 在QuartusII环境下，对设计方案进行输入。

(2) 编译、综合、适配、下载，验证计数结果。

(3) 进行时序仿真。

(4) 对计数器进行异步清0和同步使能的操作并查看结果。

设计提示：建议选“实验电路模式5”，用键8（PIO7）控制RST；用键7（PIO6）控制ENA；计数溢出COUT接发光管D8（PIO15）；OUTY是计数输出，接数码1（PIO19-PIO16，低位靠右）；时钟CLK接clock0，通过短路帽选择4Hz信号。

四、实验报告要求

(1) 写出用VHDL语言设计含异步清0和同步时钟使能的4位加法计数器的设计方案。

(2) 用QuartusII做出计数器的仿真波形,仿真波形要能反映异步清0和同步使能的情况，并与实验结果进行对比。

(3) 查看计数器的最高工作频率。

(4) 将实验过程和实验结果写进实验报告。

(5) 写出异步与同步电路的异同及电路的适用范围。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT4B IS
    PORT  (CLK : IN STD_LOGIC;
           RST : IN STD_LOGIC;
           ENA : IN STD_LOGIC;
           OUTY : OUT STD_LOGIC_VECTOR (3 DOWNTO 0) ;
          COUT : OUT STD_LOGIC    ) ;
  END CNT4B;
```

```
ARCHITECTURE behav OF CNT4B IS
    SIGNAL CQI : STD_LOGIC_VECTOR (3 DOWNTO 0) ;
BEGIN
P_REG: PROCESS (CLK, RST, ENA)
        BEGIN
         IF RST = '1' THEN   CQI <= "0000";
           ELSIF CLK'EVENT AND CLK = '1' THEN
              IF ENA = '1' THEN  CQI <= CQI + 1;
           END IF;
         END IF;
       OUTY <= CQI ;
        END PROCESS P_REG ; --进位输出
   COUT <= CQI (0) AND CQI (1) AND CQI (2) AND CQI (3) ;
END behav;
```

实验五 7段数码显示译码器设计

一、实验目的

(1) 学习使用VHDL语言设计简单组合逻辑电路。

(2) 学习使用CASE语句来描述真值表。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

(1) 用VHDL设计一个共阴数码管（图3-34）的译码电路，用case语句描述7段译码器的真值表。

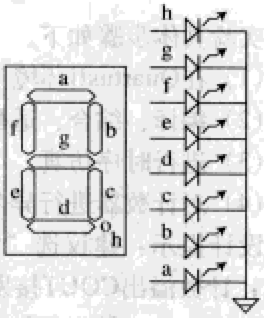


图3-34 共阴数码管及其电路

(2) 编译、综合、适配、下载，验证结果。

(3) 进行时序仿真。

(4) 设计提示：建议选实验电路模式6，用数码8显示译码输出（PIO46～PIO40），键8、键7、键6、键5四位控制输入，硬件验证译码器的工作性能。注意，在仿真中，4位输入都必须用总线方式给数据。

(5) 附加实验内容：用VHDL例化语句，按图3-35的方式，以实验四和实验五为底层元件，完成顶层文件设计，并重复以上实验过程。注意图3-35中的tmp是4位总线，led是7位总线。对于引脚锁定和实验，建议仍选实验电路模式6，用数码8显示译码输出，用键3作为时钟输入（每按2次键为1个时钟脉冲），或作为直接时钟信号clock0。

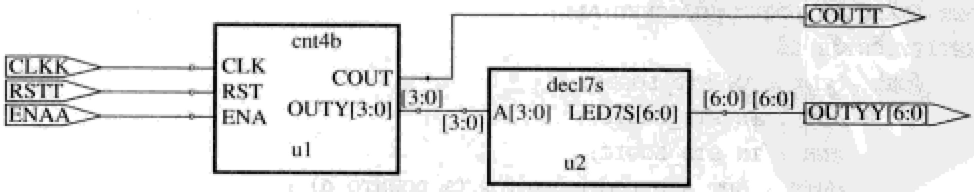


图3-35 计数器和译码器连接电路的顶层文件原理图

四、实验报告要求

（1）写出用VHDL语言设计7段译码器的设计方案。

（2）用QuartusII作出真波形，并与实验结果进行对比。

（3）将实验过程和实验结果写进实验报告。

（4）讨论语句WHEN OTHERS=>NULL的作用。对于不同的VHDL综合器，此句是否具有相同的含义和功能。

```
LIBRARY IEEE ;
 USE IEEE.STD_LOGIC_1164.ALL ;
  ENTITY DecL7S IS
    PORT  (A      : IN  STD_LOGIC_VECTOR (3 DOWNTO 0) ;
           LED7S : OUT STD_LOGIC_VECTOR (6 DOWNTO 0)  ) ;
 END ;
 ARCHITECTURE one OF DecL7S IS
 BEGIN
    PROCESS (A)
    BEGIN
        CASE  A (3 DOWNTO 0)  IS
          WHEN "0000" =>  LED7S <= "0111111" ; -- X "3F" → 0
          WHEN "0001" =>  LED7S <= "0000110" ; -- X "06" → 1
          WHEN "0010" =>  LED7S <= "1011011" ; -- X "5B" → 2
          WHEN "0011" =>  LED7S <= "1001111" ; -- X "4F" → 3
          WHEN "0100" =>  LED7S <= "1100110" ; -- X "66" → 4
          WHEN "0101" =>  LED7S <= "1101101" ; -- X "6D" → 5
          WHEN "0110" =>  LED7S <= "1111101" ; -- X "7D" → 6
          WHEN "0111" =>  LED7S <= "0000111" ; -- X "07" → 7
          WHEN "1000" =>  LED7S <= "1111111" ; -- X "7F" → 8
          WHEN "1001" =>  LED7S <= "1101111" ; -- X "6F" → 9
          WHEN "1010" =>  LED7S <= "1110111" ; -- X "77" → 10
          WHEN "1011" =>  LED7S <= "1111100" ; -- X "7C" → 11
          WHEN "1100" =>  LED7S <= "0111001" ; -- X "39" → 12
          WHEN "1101" =>  LED7S <= "1011110" ; -- X "5E" → 13
          WHEN "1110" =>  LED7S <= "1111001" ; -- X "79" → 14
          WHEN "1111" =>  LED7S <= "1110001" ; -- X "71" → 15
          WHEN OTHERS =>  NULL ;
        END CASE ;
    END PROCESS ;
 END ;

LIBRARY  IEEE;
 USE IEEE.STD_LOGIC_1164.ALL;
 ENTITY TOP_PRJ IS
   PORT  (CLKK,RSTT,ENAA :   IN STD_LOGIC;
                COUTT : OUT STD_LOGIC;
             OUTYY  : OUT STD_LOGIC_VECTOR (6 DOWNTO 0)) ;
 END ENTITY TOP_PRJ;
```

```
ARCHITECTURE fd1 OF TOP_PRJ IS
  COMPONENT  CNT4B
   PORT  (CLK : IN STD_LOGIC;
         RST : IN STD_LOGIC;
         ENA : IN STD_LOGIC;
        OUTY : OUT STD_LOGIC_VECTOR (3 DOWNTO 0) ;
        COUT : OUT STD_LOGIC      ) ;
  END COMPONENT;
    COMPONENT DECL7S
   PORT  (   A   : IN  STD_LOGIC_VECTOR (3 DOWNTO 0) ;
          LED7S  : OUT STD_LOGIC_VECTOR (6 DOWNTO 0)  ) ;
  END COMPONENT;
 SIGNAL AB   :  STD_LOGIC_VECTOR (3 DOWNTO 0) ;
 BEGIN
  u1 : CNT4B PORT MAP (CLK=>CLKK,RST=>RSTT,ENA=>ENAA,OUTY=>AB,COUT=>COUTT) ;
  u2 : DECL7S PORT MAP (A=>AB, LED7S=>OUTYY) ;
 END ARCHITECTURE fd1;;
```

实验六　巴克码信号发生器设计

一、实验目的

（1）学习使用VHDL语言设计时序电路。

（2）了解设计周期性的序列信号发生器的方法。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

（1）用VHDL设计一个序列信号发生器，要求能自动产生周期性的1110010序列。

（2）在QuartusII环境下，对设计方案进行输入。

（3）编译、综合、适配和下载，验证结果。

（4）进行时序仿真。

设计提示：建议选实验电路模式4，键7（PIO9）作为时钟输入，PIO10作为周期序列信号的输出。

四、实验报告要求

（1）写出用VHDL语言设计序列信号发生器的设计方案。

（2）用QuartusII做出真波形,并与实验结果进行对比。

（3）将实验过程和实验结果写进实验报告。

（4）如果用状态机来设计，应如何改写VHDL程序。

（5）如果要求设计一个伪随机信号发生器，应如何进行设计。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity sigout is port (
```

```
    clk : in std_logic;
    cout : out std_logic) ;
end sigout;
architecture one of sigout is
    signal q : std_logic_vector (2 downto 0) ;
begin
    process (clk)
    begin
        if rising_edge (clk) then
            if q = "110" then
                q <= "000" ;
            else
                q <= q + 1;
            end if;
        end if;
    end process;
    process (q)
    begin
        case q is
            when "000" => cout <= '1' ;
            when "001" => cout <= '1' ;
            when "010" => cout <= '1' ;
            when "011" => cout <= '0' ;
            when "100" => cout <= '0' ;
            when "101" => cout <= '1' ;
            when "110" => cout <= '0' ;
            when others => cout <= '-' ;
        end case;
    end process;
end one;
```

实验七　多功能数字钟设计

一、实验目的

(1) 学习使用VHDL设计较为复杂的逻辑电路。

(2) 掌握自顶向下的设计方法。

(3) 设计一个具有复位、整点响铃和时间设置功能的数字钟。

二、实验设备与器材

GW-PK2 EDA实验箱一台。

三、实验内容及实验步骤

(1) 用VHDL设计一个数字钟，要求具有复位、整点响铃和时间设置功能。

(2) 采用自顶向下的方法进行设计（多功能时钟的原理框图如图3-36所示，各功能模块采用VHDL设计。其中cnt60、cnt24为计数模块，ring为响铃控制，clk_ring为分频器，

encode14为预置数的时分秒控制。）

（3）编译、综合、适配和下载，验证结果。

（4）进行时序仿真。

设计提示：建议选实验电路模式7进行设计。

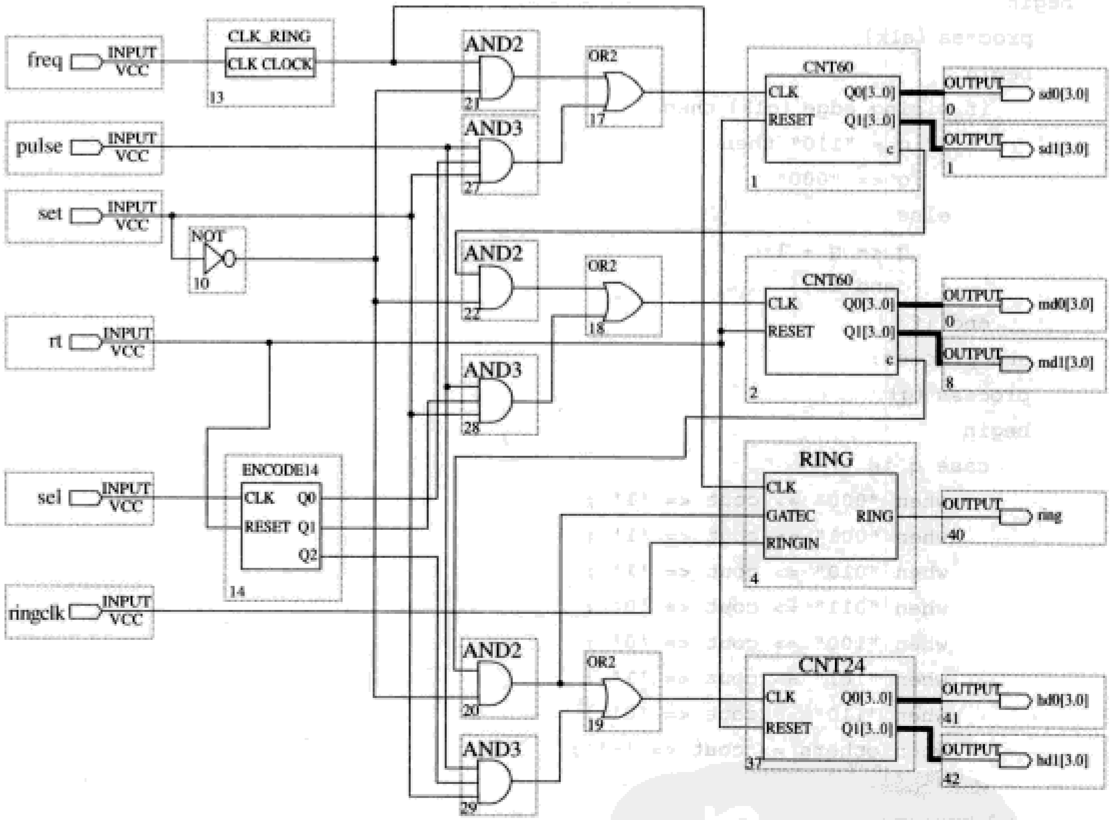


图3-36　多功能时钟的原理框图

四、实验报告要求

（1）写出用VHDL语言设计数字钟的设计方案，并详细说明其工作原理。

（2）用QuartusII做出真波形,并与实验结果进行对比。

（3）将实验过程和实验结果写进实验报告。

（4）如果要求响铃的次数与当时的时间有关，应如何修改相应模块。

（5）如果要求数字钟具有闹钟的功能，应如何修改设计。

```
--分频器
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity clk_ring is
port (
```

```
        clk:in std_logic;
        clock:out std_logic) ;--1Hz
输出
  end;
  architecture one of clk_ring is
  signal tmp:std_logic_vector (9 downto 0) ;
begin
process (clk)
begin
  if rising_edge (clk) then
     if  (tmp=3) then
        tmp <= (others => '0 ') ;
     else
        tmp <= tmp+1;
    end if;
  end if;
end process;

Clock <= tmp (1) ;--正负脉宽相等，用于控制响铃
end one;

--预置选择
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity encode14 is
port (clk,reset:in std_logic;
        q0,q1,q2:out std_logic) ;
end;

architecture one of encode14 is
signal tmp:std_logic_vector (1 downto 0) ;
Begin

process (clk,reset)
begin
if reset='1' then
   tmp <= "00";
elsif rising_edge (clk) then
   tmp <= tmp+1;
end if;
end process;

q0<= (not tmp (0) ) and  (not tmp (1) ) ;
q1<=tmp (0) and  (not tmp (1) ) ;
q2<= (not tmp (0) ) and tmp (1) ;
end one;

--60进制计数，2位十进制输出
library ieee;
```

```
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity cnt60 is
port (clk,reset:in std_logic;
q0,q1:buffer std_logic_vector (3 downto 0) ;
c:out std_logic) ;
end;

architecture one of cnt60 is
begin
process (clk,reset)
begin
if reset='1' then
   q0<="0000";
   q1<="0000";
elsif rising_edge (clk) then
   if  (q0=9) and  (q1=5) then
      q0<="0000";
      q1<="0000";
   elsif  (q0=9) then
      q0<="0000";
      q1<=q1+"0001";
   else
      q0<=q0+"0001";
   end if;
end if;
end process;
process (clk)
begin
if rising_edge (clk) then
if  (q0=9) and  (q1=5) then
   c<= '1' ;
else
   c<='0';
end if;
end if;
end process;
end one;

--24进制计数，2位十进制输出

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity cnt24 is
        port (clk,reset:in std_logic;
```

```
          q0,q1:buffer std_logic_vector (3 downto 0)
        ) ;
end;
architecture one of cnt24 is
begin
process (clk,reset)
begin
if reset='1' then
   q0<="0000";
   q1<="0000";
elsif rising_edge (clk) then
   if  (q0=3) and  (q1=2) then
      q0<="0000";
      q1<="0000";
   elsif  (q0=9) then
      q0<="0000";
      q1<=q1+"0001";
   else
      q0<=q0+"0001";
   end if;
end if;
end process;
end one;

library ieee; --响铃控制
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity ring is
port (
          clk,gatec,ringin:in std_logic;
          ring:out std_logic
        );
end;
architecture one of ring is
signal tmp:std_logic_vector (1 downto 0) ;
else '0';
Begin
process (clk,gatec)

begin
if gatec='0' then
   tmp<="00";
elsif rising_edge (clk) then
   if (tmp<2) then
      tmp<=tmp+1;
```

```
        else
            tmp<=tmp;
        end if;
      end if;
    end process;
ring<=ringin and clk and gatec when tmp<2
      end one;
```

实验八　状态机实现的ADC0809采样控制电路

一、实验目的

学习用状态机实现A/D转换器ADC0809的采样控制电路。

二、实验原理

ADC0809是CMOS的8位A/D转换器，片内有8路模拟开关，可控制8个模拟量中的一个进入转换器中。ADC0809的分辨率为8位，转换时间约100μs，含锁存控制的8路多路开关，输出有三态缓冲器控制，单5V电源供电。

主要控制信号说明如下。如图3-37所示，START是转换启动信号，高电平有效；ALE是3位通道选择地址（ADDC、ADDB、ADDA）信号的锁存信号。当模拟量送至某一输入端（如IN1或IN2等），由3位地址信号选择，而地址信号由ALE锁存；EOC是转换情况状态信号（类似于AD574的STATUS），当启动转换约100μs后，EOC产生一个负脉冲，以示转换结束；在EOC的上升沿后，若使输出使能信号OE为高电平，则控制打开三态缓冲器，把转换好的8位数据结果输至数据总线。至此ADC0809的一次转换结束了。

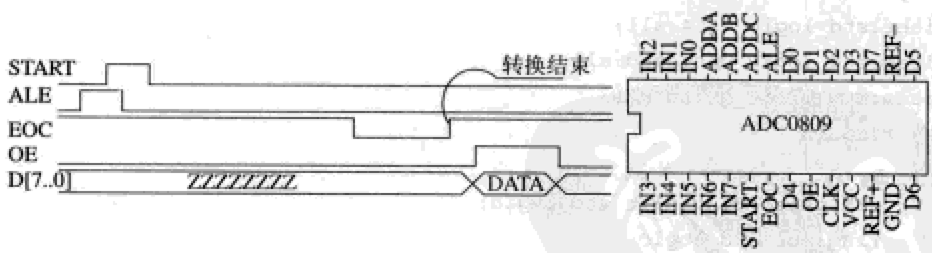


图3-37　ADC0809工作时序

三、实验设备与器材

GW-PK2 EDA实验箱一台。

四、实验内容及实验步骤

（1）利用QuartusII对设计文件进行文本编辑输入和仿真测试，给出仿真波形。

（2）进行引脚锁定并进行测试，硬件验证电路对ADC0809的控制功能。

（3）选择模式5，建议引脚锁定为START接PIO34，OE接PIO35，EOC接PIO8，ALE接PIO33，状态机时钟CLK接clock0（PIN2，可选“65536Hz”或更高），ADDA接PIO32（ADDB和ADDC都接GND），ADC0809的8位输出数据线接PIO23～PIO16，锁存输出Q显示于数码8/数码7（PIO47～PIO40），拨码开关选择“转换结束”和“A/D使能”。下载目标文件后，可用螺丝刀旋转实验系统左下角的电位器，以便为ADC0809提供变化的待测模拟信号，这时数码管8和7将显示ADC0809采样输出和被锁存的数字值（16进制）。

五、实验报告要求

(1) 写出用VHDL语言设计A/D控制器的设计方案，并详细说明其工作原理。

(2) 用QuartusII做出真波形,并与实验结果进行对比。

(3) 将实验过程和实验结果写进实验报告。

(4) 如果采用状态位直接输出的方式设计状态机控制ADC0809，应如何修改设计。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY ADCINT IS
    PORT  (D   : IN STD_LOGIC_VECTOR (7 DOWNTO 0) ; --0809的8位转换数据输出
                              CLK ,EOC  : IN STD_LOGIC; --CLK是转换工作时钟
     LOCK1, ALE, START, OE, ADDA   : OUT STD_LOGIC;
         Q   : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)) ;
END ADCINT;
ARCHITECTURE behav OF ADCINT IS
TYPE states IS  (st0, st1, st2, st3,st4,st5,st6) ; --定义各状态子类型
  SIGNAL current_state, next_state: states :=st0 ;
  SIGNAL REGL        : STD_LOGIC_VECTOR (7 DOWNTO 0) ;
  SIGNAL LOCK        : STD_LOGIC; -- 转换后数据输出锁存时钟信号
 BEGIN
    ADDA <= '1'; LOCK1 <=LOCK;
  PRO: PROCESS (current_state,EOC)  BEGIN  --规定各状态转换方式
   CASE current_state IS
   WHEN st0 => ALE<='0';START<='0';OE<='0';LOCK<='0' ;next_state <= st1;
   WHEN st1 => ALE<='1';START<='0';OE<='0';LOCK<='0' ;next_state <= st2;
   WHEN st2 => ALE<='0';START<='1';OE<='0';LOCK<='0' ;next_state <= st3;
   WHEN st3 => ALE<='0';START<='0';OE<='0';LOCK<='0';
     IF  (EOC='1') THEN next_state <= st3;     --测试EOC的下降沿
 ELSE next_state <= st4;
     END IF ;
   WHEN st4=> ALE<='0';START<='0';OE<='0';LOCK<='0';
     IF  (EOC='0') THEN next_state <= st4; --测试EOC的上升沿，=1表明转换结束
    ELSE next_state <= st5;                       --继续等待
     END IF ;
   WHEN st5=> ALE<='0';START<='0';OE<='1';LOCK<='0';next_state <= st6;
   WHEN st6=> ALE<='0';START<='0';OE<='1';LOCK<='1';next_state <= st0;
   WHEN OTHERS => ALE<='0';START<='0';OE<='0';LOCK<='0';next_state <= st0;
   END CASE ;
  END PROCESS PRO ;
    PROCESS  (CLK)
      BEGIN
        IF  (CLK'EVENT AND CLK='1')  THEN
           current_state <= next_state; -- 在时钟上升沿，转换至下一状态
        END IF;
    END PROCESS;   -- 由信号current_state将当前状态值带出此进程，进入进程PRO
```

```
    PROCESS  (LOCK) -- 此进程中，在LOCK的上升沿，将转换好的数据锁入
    BEGIN
    IF LOCK='1' AND LOCK'EVENT THEN   REGL <= D ;
    END IF;
   END PROCESS ;
         Q <= REGL;
END behav;
```

实验九 波形发生与扫频信号发生器电路设计

一、实验目的

学习用VHDL设计波形发生器和扫频信号发生器，掌握FPGA对D/A的接口和控制技术，学会LPM_ROM在波形发生器设计中的实用方法。

二、实验原理

如图3-38所示，完整的波形发生器由4部分组成。

（1）FPGA中的波形发生器控制电路，通过外来控制信号和高速时钟信号，向波形数据ROM发出地址信号，输出波形的频率由发出的地址信号的速度决定。当以固定频率扫描输出地址时，模拟输出波形是固定频率；当以周期性时变方式扫描输出地址时，模拟输出波形为扫频信号。

（2）波形数据ROM中存有发生器的波形数据，如正弦波或三角波数据。当接收来自FPGA的地址信号后，将从数据线输出相应的波形数据，地址变化得越快，则输出数据的速度越快，从而使D/A输出的模拟信号的变化速度越快。波形数据ROM可以由多种方式实现，如在FPGA外面外接普通ROM，由逻辑方式在FPGA中实现（如本例），或由FPGA中的EAB模块担当，如利用LPM_ROM实现。相比之下，第1种方式的容量最大，但速度最慢；第2种方式容量最小，但速度最快；第3种方式则兼顾了速度和ROM波形容量两方面的因素。

（3）D/A转换器负责将ROM输出的数据转换成模拟信号。

（4）D/A转换后的信号经低通滤波电路后输出。

本例中的正弦波波形数据由64个点构成，此数据经DAC0832，并经滤波器后，可在示波器上观察到光滑的正弦波（若接精密基准电压，可得到更为清晰的正弦波形）。

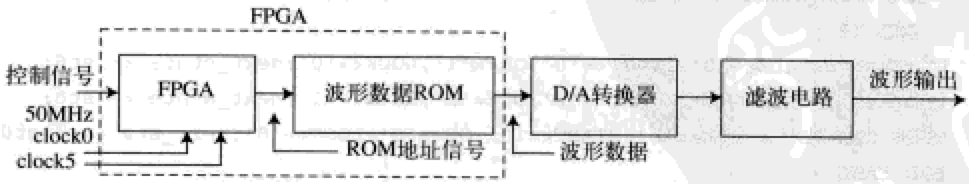


图3-38 波形发生器电路系统结构图

三、实验设备与器材

（1）GW-PK2 EDA实验箱一台。

（2）示波器一台。

四、实验内容及实验步骤

（1）根据示例程序及以上的设计原理，完成波形发生器和扫频信号源的设计、仿真测试

及实验系统上的硬件测试。

（2）硬件实验中注意DAC0832及滤波电路需接+/–12V电压。然后将实验系统左下角选择插针处用短路帽短路"D/A直通"，而"滤波1"、"滤波0"处通过短路或不接短路帽达到不同的滤波方式。将示波器的地与EDA实验系统的地相接，信号端与"AOUT"信号输出端相接；建议CLK接clock0，由此接50MHz频率，此频率扫描波形数据；CLK1接clock5，由此接"1024Hz"，此频率决定扫频速度；选电路模式1；KK接键8，当为高电平时，正弦波固定频率输出，11位输入数据DATA由键3、键2和键1控制，信号源的输出频率由此3键输入的12位二进制数决定，数值越大，输出频率越高； "FD0"时为最高频率；键8低电平时，正弦波扫频输出，扫频速度由clock5的频率决定。输向0832的8位数据由DD输出。

（3）在本例中插入一个LPM_ROM，将原例中的波形数据放在内部ROM中。必要时增加波形点数，以便低频输出时，仍保持良好波形。波形数据可由其他方式自动生成，然后重复以上的测试和硬件实验。

五、实验报告要求

（1）写出用VHDL语言设计波形发生器的方案，并详细说明其工作原理。

（2）用QuartusII做出真波形,并与实验结果进行对比。

（3）将实验过程和实验结果写进实验报告。

（4）如果CLK的输入频率是50MHz，ROM中一个周期的正弦波数据是128个，要求输出的正弦波频率不低于150kHz，DAC0832是否能适应此项工作？为什么？

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY DAC IS
    PORT  (CLK,CLK1,KK : IN STD_LOGIC;
                DATA    : IN STD_LOGIC_VECTOR (11 DOWNTO 0) ;
                    DD : OUT INTEGER RANGE 255 DOWNTO 0 ) ;
END;
ARCHITECTURE DACC OF DAC IS
    SIGNAL Q         : INTEGER RANGE 63 DOWNTO 0 ;
    SIGNAL D         : INTEGER RANGE 255 DOWNTO 0 ;
    SIGNAL FSS       : STD_LOGIC   ;
    SIGNAL COUNT12,DATA2,DATA1  : STD_LOGIC_VECTOR (11 DOWNTO 0) ;
BEGIN
    PROCESS (FSS)
    BEGIN
        IF (FSS' EVENT AND FSS = '1' ) THEN   Q <= Q + 1;
        END IF;
    END PROCESS;
  PROCESS ( Q )
    BEGIN
 CASE Q IS
WHEN 00=> D<=255; WHEN 01=> D<=254; WHEN 02=> D<=252; WHEN 03=> D<=249;
WHEN 04=> D<=245; WHEN 05=> D<=239; WHEN 06=> D<=233; WHEN 07=> D<=225;
WHEN 08=> D<=217; WHEN 09=> D<=207; WHEN 10=> D<=197; WHEN 11=> D<=186;
```

```
WHEN 12=> D<=174; WHEN 13=> D<=162; WHEN 14=> D<=150; WHEN 15=> D<=137;
WHEN 16=> D<=124; WHEN 17=> D<=112; WHEN 18=> D<= 99; WHEN 19=> D<= 87;
WHEN 20=> D<= 75; WHEN 21=> D<= 64; WHEN 22=> D<= 53; WHEN 23=> D<= 43;
WHEN 24=> D<= 34; WHEN 25=> D<= 26; WHEN 26=> D<= 19; WHEN 27=> D<= 13;
WHEN 28=> D<=  8; WHEN 29=> D<=  4; WHEN 30=> D<=  1; WHEN 31=> D<=  0;
WHEN 32=> D<=  0; WHEN 33=> D<=  1; WHEN 34=> D<=  4; WHEN 35=> D<=  8;
WHEN 36=> D<= 13; WHEN 37=> D<= 19; WHEN 38=> D<= 26; WHEN 39=> D<= 34;
WHEN 40=> D<= 43; WHEN 41=> D<= 53; WHEN 42=> D<= 64; WHEN 43=> D<= 75;
WHEN 44=> D<= 87; WHEN 45=> D<= 99; WHEN 46=> D<=112; WHEN 47=> D<=124;
WHEN 48=> D<=137; WHEN 49=> D<=150; WHEN 50=> D<=162; WHEN 51=> D<=174;
WHEN 52=> D<=186; WHEN 53=> D<=197; WHEN 54=> D<=207; WHEN 55=> D<=217;
WHEN 56=> D<=225; WHEN 57=> D<=233; WHEN 58=> D<=239; WHEN 59=> D<=245;
WHEN 60=> D<=249; WHEN 61=> D<=252; WHEN 62=> D<=254; WHEN 63=> D<=255;
WHEN OTHERS => NULL ;
END CASE;
END PROCESS;
  DD <= D ;
   PROCESS (CLK, DATA)
     BEGIN
      IF CLK' EVENT AND CLK = '1' THEN
        IF COUNT12 = "111111100000" THEN COUNT12 <= DATA1; FSS <= '1' ;
           ELSE  COUNT12 <= COUNT12 + 1; FSS <= '0' ;
        END IF;
      END IF;
  END PROCESS;
  DATA1 <= DATA   WHEN KK = '1' ELSE
             DATA2  WHEN KK = '0' ELSE DATA2 ;
  PROCESS (CLK1)
    BEGIN
        IF (CLK1' EVENT AND CLK1 = '1' ) THEN DATA2 <= DATA2 + 1;
        END IF;
    END PROCESS;
 END;
```

3.3 可编程器件课程设计

课程设计一　自动售邮票机

1. 设计说明

本课程设计的任务是设计一个自动售邮票机。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog）。

（c）对设计电路进行软件仿真与硬件测试。

3. 设计要求

（a）机器能自动识别硬币金额，最大为1元，最小为1角。购票者可选择的邮票面值有1元和5角两种，每次只能售出一枚邮票。

（b）购票时先选择邮票面值后投币，当投入的硬币总金额达到或超过邮票面值时，机器应发出指示并拒收继续投入的硬币。

（c）购票者投币后，按确认键，机器将发出邮票和找零硬币。若所投金额不足，则发出欠资指示。在欠资情况下，购票者可以继续投币购票，也可按取消键，则机器将退出所投入的全部金额。

（d）找零或退币时，按从大到小原则处理，即总金额超过1元时，应找出1元硬币，金额低于1元但超过5角时，应找出5角硬币，不足5角时，方才找出1角硬币。

4. 设计提示

根据设计要求画出状态转移图，并用VHDL或Verilog来实现此状态转移图。

课程设计二　8位二进制乘法电路

1. 设计说明

纯组合逻辑构成的乘法器虽然工作速度比较快，但过于占用硬件资源，难以实现宽位乘法器。本课程设计的任务就是设计一个由8位加法器构成的以时序逻辑方式工作的8位乘法器。其特点是占用硬件资源少。其工作原理是，乘法通过逐项移位相加原理来实现，从被乘数的最低位开始，若为1，则乘数左移后与上一次的和相加；若为0，左移后以全零相加，直至被乘数的最高位。图3-39清楚地说明了此乘法器的工作原理。

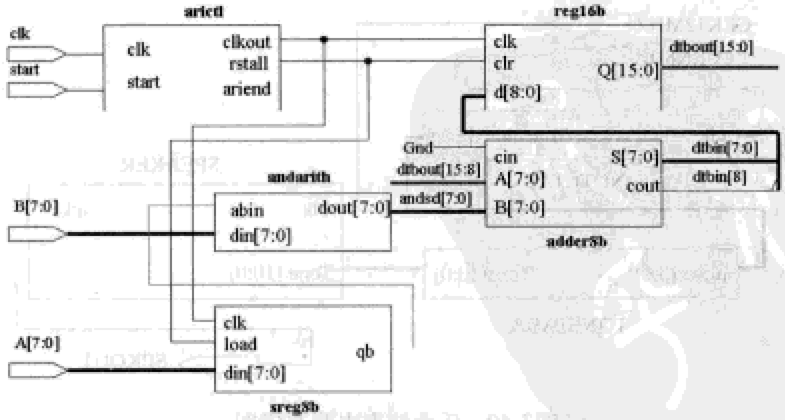


图3-39　8×8位乘法器电路原理图

2. 设计内容

（a）进行需求分析，确定总体框架。

（b）分模块具体设计，给出源文件（VHDL或Verilog）。

（c）对设计电路进行软件仿真与硬件测试。

3. 设计要求

利用逐项移位相加原理来实现8位乘法器。

4. 设计提示

利用8位加法器与移位寄存器参照图3-52来实现8位乘法器。

课程设计三　乐曲演奏电路设计

1. 设计说明

与利用微处理器来实现乐曲演奏相比，以纯硬件实现的电路的逻辑要复杂得多，本课程设计的任务是利用EDA工具和硬件描述语言来设计一个可以自动播放乐曲的电路。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行软件仿真与硬件测试。

3. 设计要求

(a) 能够自动播放设计好的乐曲。

(b) 至少能选择播放二个以上的乐曲。

4. 设计提示

组成乐曲的每个音符的发音频率及其持续的时间是乐曲能连续演奏的两个基本要素。所以，硬件电路只要能够控制发音频率及持续的时间就可以实现乐曲的演奏。乐曲演奏电路逻辑图如图3-40所示，其中，NOTETABS模块用于存贮乐曲的曲谱和控制乐曲演奏的节拍；TONETABA模块根据乐谱输出分频预置数，供SPEAKER模块使用，音符与频率的关系参照表2-15；SPEAKER实质上就是一个分频器，它通过给定的分频预置数进行分频。

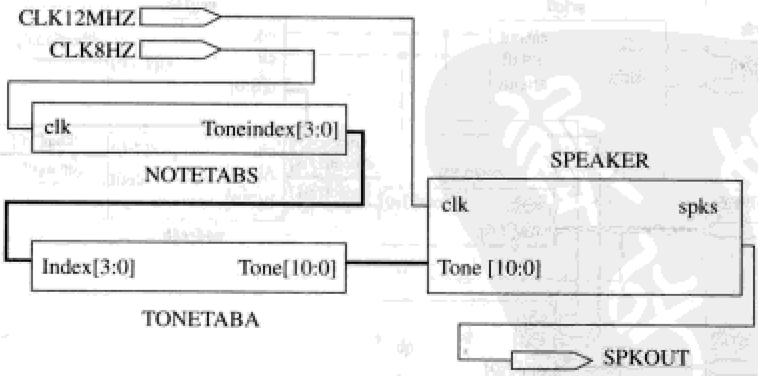


图3-40　乐曲演奏电路逻辑图

课程设计四　VGA显示器彩色信号发生器设计

1. 设计说明

利用可编程器件FPGA实现的彩色显示控制器在工业现场有许多实际应用。本课程设计的任务是设计一个基于FPGA器件的VGA控制器，该控制器可以实现显示器的条纹信号和方

格信号。

2. 设计内容

(a) 掌握电视信号的基础知识。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

(a) 在ModelSim环境下实现控制器的仿真。

(b) 在QuartusII环境下实现对VGA信号发生器的设计。

(c) 在VGA显示器上显示出所设计的条纹信号和方格信号。

4. 设计提示

(a) 了解VGA信号中行扫描和场扫描信号的时序关系。

(b) 了解VGA行场正程、消隐的时间以及行同步、场同步的位置及时间宽度。

(c) 利用分频器来实现设计。

课程设计五　8位十进制频率计设计

1. 设计说明

频率测量的基本原理是计算每秒钟内待测信号的脉冲个数。本课程设计的任务是利用这一原理设计实现一个8位十进制频率计。

2. 设计内容

(a) 掌握频率计的基本工作原理。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

(a) 设计实现一个8位十进制频率计。

(b) 测量频率范围为10Hz～50MHz；

4. 设计提示

利用8个十进制计数器、分频器和32位寄存器，结合测频控制电路实现。测频控制电路的工作时序如图3-41所示，利用分频器将1Hz的信号分频为正负脉宽都是1s的信号cnt_en；将cnt_en信号取反，得到load信号；将1Hz信号与cnt_en信号通过或非门得到rst_cnt信号。当cnt_en为高电平时，十进制计数器开始计数，cnt_en为低电平时计数终止。当load信号为上升沿时，将计数的结果锁存到寄存器，供数码管显示；当rst_cnt信号为上升沿时，计数器清零，为下一次计数作准备。

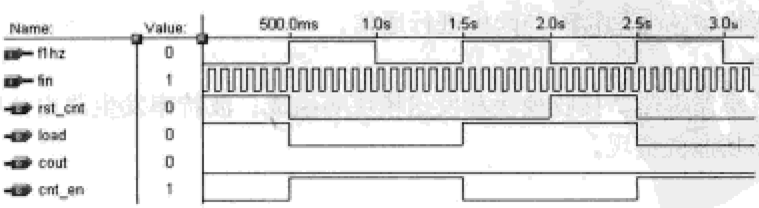


图3-41　测频控制信号发生器的工作时序

课程设计六 简易波形发生器设计

1. 设计说明

本课程设计的任务是设计一台简易波形发生器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

(a) 波形发生器能够产生正弦波、方波或三角波、AM、FM调制信号和二进制PSK、ASK信号。

(b) 在1kHz~10MHz的范围内以100Hz的步进调整。

4. 设计提示

采用DDS原理来实现此波形发生器。如图3-42所示。

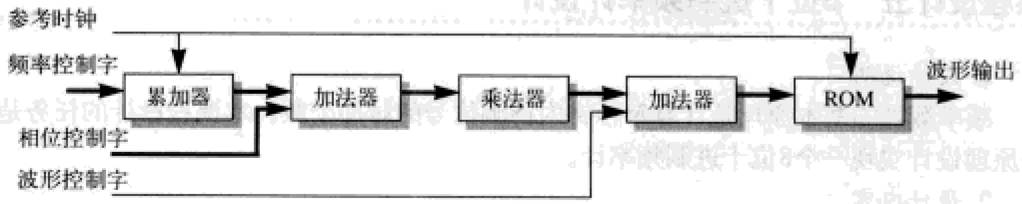


图3-42 DDS系统模型

课程设计七 RS232通信接口

1. 设计说明

RS232异步通信在智能仪表、工业自动控制系统和通信设备中有着广泛的应用。本课程设计的任务是利用硬件描述语言在FPGA中实现RS232模块。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

(a) 在FPGA中实现RS232异步通信模块。

(b) 利用所设计的通信接口与pc机进行通信。

4. 设计提示

系统由波特率发生器、接收模块和发送模块等组成。波特率发生器由分频器实现，接收和发送模块利用状态机实现。

课程设计八 串行二进制码与BCD码变换电路

1. 设计说明

串行二进制码/BCD码转换电路原理如图3-43所示。二进制码由右端串行送入，高位在前。当每个变换（4bit）单元中数据大于或等于5时，如再有新的数据送入，则应利用修正网络加以修正。方法是将原有数据加3（即0011）后左移一位，例如变换单元内现有数据为0110,如需再移入一位数据（假设为1）,则将原来数据0110加0011后得1001,然后左移1位，并将输入数据移入得XXX1，0011。

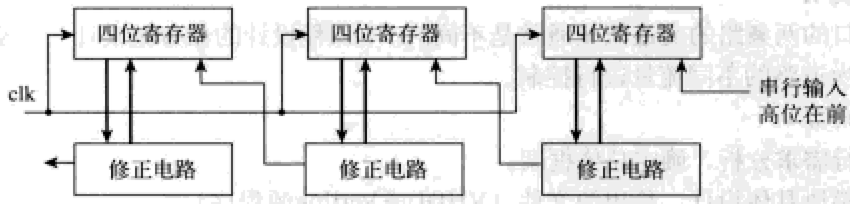


图3-43 串行二进制码/BCD码转换电路

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

能对串行输入的8bit二进制码进行转换，使输出结果为3位BCD码，输出可以用LED显示，也可经译码用数码管显示（串行送入的二进制码高位在前）。

4. 设计提示

应先设计修正网络。修正网络可以用时序电路实现“过5加3”功能，也可以用全加器实现。

课程设计九 智能裁决器

1. 设计说明

本课程设计的任务是设计一台可供八人抢答，时间可预置的智力竞赛抢答装置。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

(a) 智能裁决器可供八人进行抢答。

(b) 计时电路是一个能预置定时时间倒计时计数器，它能预置30s、60s或90s。

(c) 按下“启动按钮”，计时器按照预置好的抢答时间开始倒计时，抢答者编号显示为0。

(d) 抢答开始后，在规定时间内，有人抢答，则显示抢答者的编号，同时计数器停止计数，并稳定显示当前抢答的时间，报警电路也产生报警。

(e) 计数器数码管显示“0”以后，不能再进行抢答。

4. 设计提示

使用优先编码器与计数器进行设计。

课程设计十　交通信号灯

1. 设计说明

十字路口的两条路的交通流量通常是不同的。本课程设计的任务是设计一个交通控制灯，能够根据主次道路的不同流量进行控制。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行仿真与测试。

3. 设计要求

该交通灯控制器用于主干道与支道公路的交叉路口，要求是优先保证主干道的畅通，因此，设计要求如下。

(a) 平时处于“主干道绿灯，支道红灯”状态，只有在支道有车辆要穿过主干道时，才将交通灯切向“主干道红灯，支道绿灯”，一旦支道无车辆通过路口，交通灯又回到“主干道绿灯，支道红灯”的状态。

(b) 主干道每次通行的时间不得短于1min，支路每次通行的时间不得长于20s，而这两个状态交换过程中出现“主干道黄等，支道红灯”和“主干道红灯，支道黄灯”的状态，持续时间都为4s。

4. 设计提示

(a) 用状态机来设计实现。

(b) 可用计数器分频来实现状态之间的切换。

课程设计十一　FIR滤波器设计

1. 设计说明

有限冲激响应（FIR）数字滤波器和无限冲激响应（IIR）数字滤波器广泛应用于数字信号处理系统中。IIR数字滤波器方便简单，但其相位的非线性，要求采用全通网络进行相位校正，且稳定性难以保障。FIR滤波器具有很好的线性相位特性，使得它越来越受到广泛的重视。本课程设计的任务就是设计一个FIR滤波器。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行软件仿真与硬件测试。

(d) 分析误差。

3. 设计要求

(a) 设计并实现一个11阶8位的FIR滤波器。

(b) 截止频率为1kHz，采样频率为10kHz。

4. 设计提示

(a) 利用Matlab计算出FIR滤波器的各项参数。

(b) 滤波器由乘加模块和寄存器实现。

课程设计十二　8051与FPGA接口逻辑设计

1. 设计说明

在一个实际的数字系统中经常会同时用到单片机与FPGA。因此需要设计单片机与FPGA之间的接口电路，以及编写相应的程序完成二者之间的数据传输。本课程设计的任务是要求通过总线扩展方式实现单片机与FPGA之间的通信。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行软件仿真与硬件测试。

3. 设计要求

(a) 掌握单片机的基础知识。

(b) 编写单片机和FPGA按总线方式进行传输的相关程序。

(c) 单片机利用FPGA接口程序实现对FPGA片内RAM的读写。

4. 设计提示

(a) 详细了解单片机读写外部存贮器的工作时序。

(b) 注意数据总线采用双向端口设计。

课程设计十三　I²C总线控制器的VHDL设计及实现

1. 设计说明

I²C是一种较为常用的串行接口标准，具有协议完善、通信效率高、支持芯片较多和占用I/O线较少等优点。现在，I²C总线已经成为一个国际标准，应用涉及家电、通信和控制等众多领域。本课程设计正是利用硬件描述语言编写I²C总线控制接口，利用此I²C接口实现与具有I²C接口的EEPROM的通信。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行软件仿真与硬件测试。

3. 设计要求

(a) 掌握VHDL语言，写出I²C接口程序。

(b) 在ModelSim环境下实现I²C接口程序的仿真。

(c) 在Quartus环境下实现I²C接口程序的综合。

(d) 在EDA实验箱上利用所设计的接口程序，实现与eeprom（AT24C256）之间的通信。

4. 设计提示

掌握具有I^2C总线接口的AT24C256的工作时序，利用状态机来实现FPGA与AT24C256之间的通信。

课程设计十四　设计一个信号频谱分析仪

1. 设计说明

设计、制作一个可分析信号频率成分，并可测量正弦信号失真度的频谱分析仪。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行软件仿真与硬件测试。

3. 设计要求

(a) 输入信号电压范围（峰峰值）为100mV～5V。

(b) 输入信号包含的频率成分范围为200Hz～10kHz。

(c) 频率分辨率为100Hz（可正确测量被测信号中频差不小于100Hz的频率分量的功率值）。

(d) 测量被测正弦信号的失真度。

4. 设计提示

(a) 可采用8051单片机完成键盘输入与结果显示，用FPGA完成FFT运算。

(b) 正弦波的失真度是指所有谐波能量之和与基波能量之比的平方根。

课程设计十五　集成电路芯片测试仪

1. 设计说明

设计制作一个集成电路芯片测试仪（图3-44），能对常用的74系列逻辑芯片进行逻辑功能测试，以确定芯片的好坏和型号。

2. 设计内容

(a) 进行需求分析，确定总体框架。

(b) 分模块具体设计，给出源文件（VHDL或Verilog源程序）。

(c) 对设计电路进行软件仿真与硬件测试。

3. 设计要求

(a) 通过键盘输入型号，可以对74系列的00/02/04/08/10/11/20/21/27/30十种组合逻辑芯片进行逻辑功能测试，确定其功能正确性。

(b) 通过键盘输入引脚特性，可以确定上述74系列的组合逻辑芯片的型号。

(c) 显示上述芯片的逻辑符号和逻辑表达式。

4. 设计提示

可采用8051单片机结合FPGA完成设计要求。其中单片机用于键盘输入和测试结果显示等功能，FPGA用于集成电路的测试，并将测试结果返回给单片机。

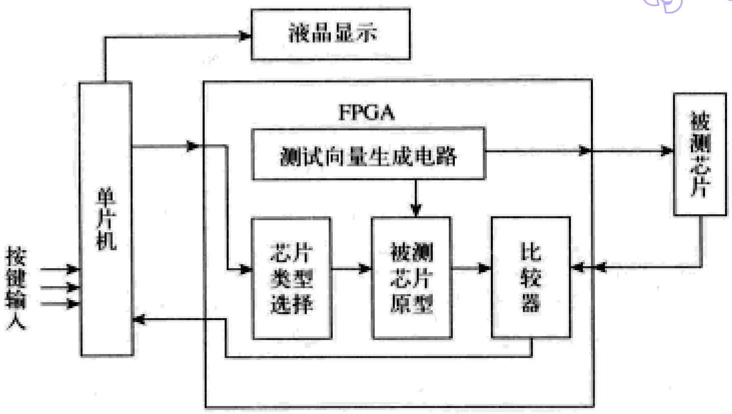
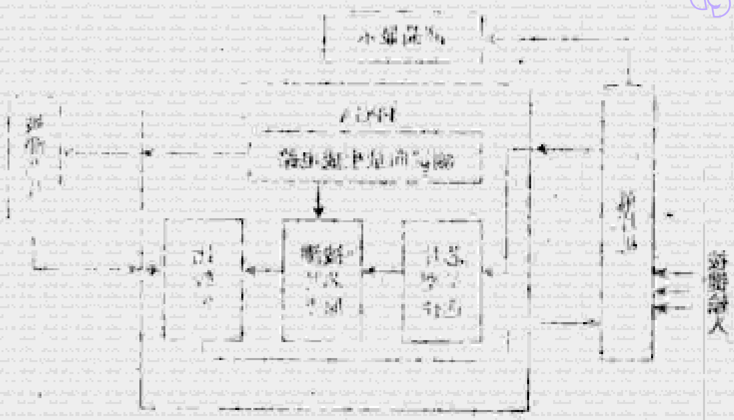
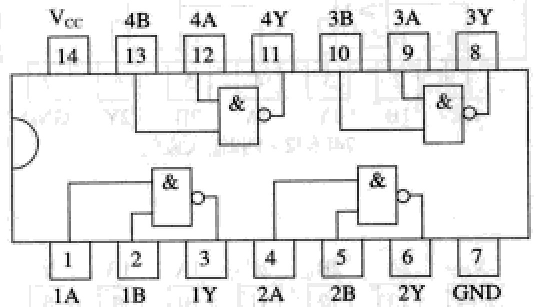


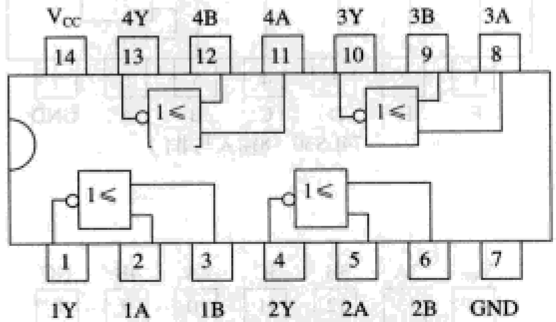
图3-44 集成电路测试框



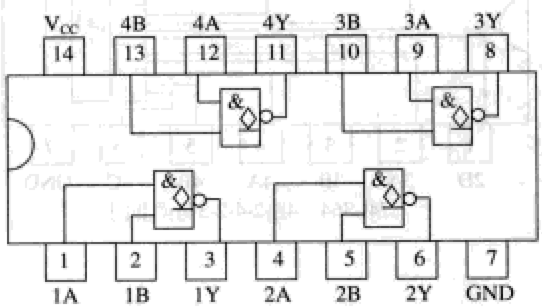
附录1　部分集成电路引脚排列图



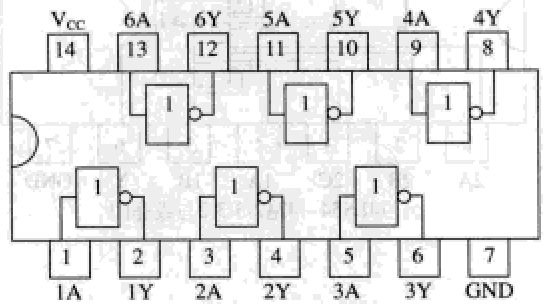
74LS00　四2输入与非门



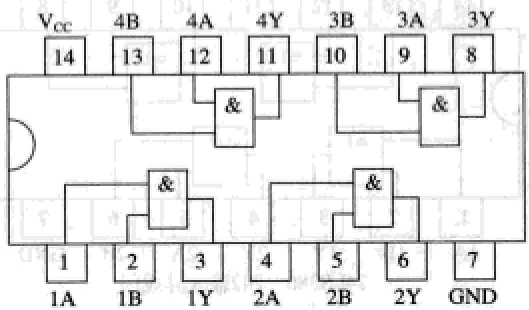
74LS28 74LS02　四2输入或非门



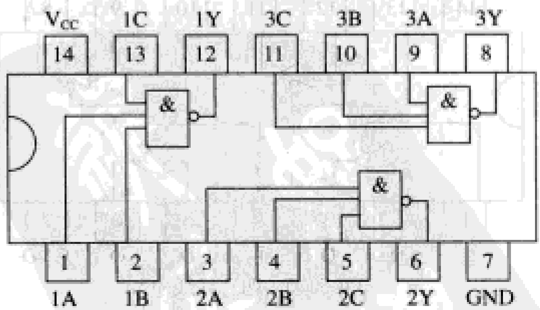
74LS03　四2输入与非门（OC）



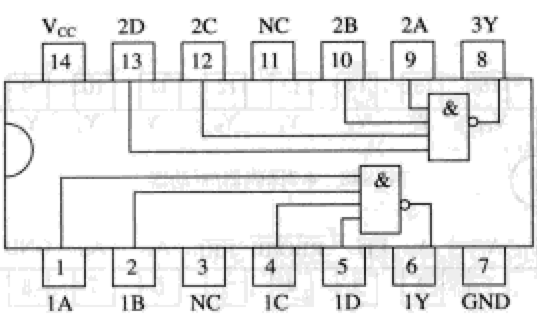
74LS04　六反相器



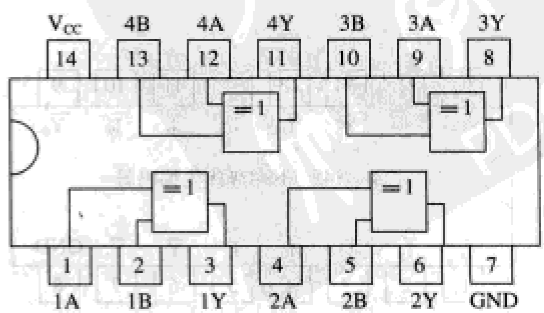
74LS08　四2输入与门



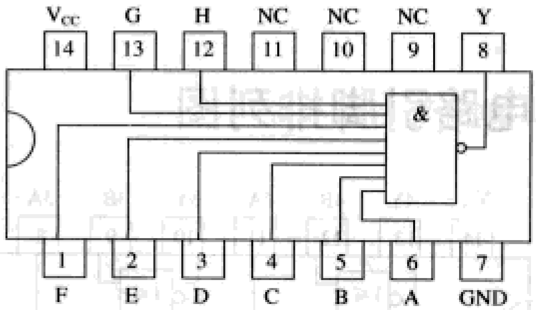
74LS10　三3输入与非门



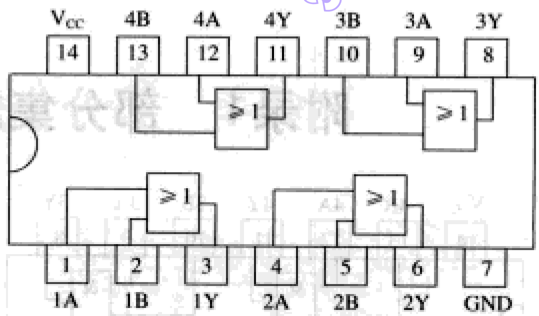
74LS20　双4输入与非门



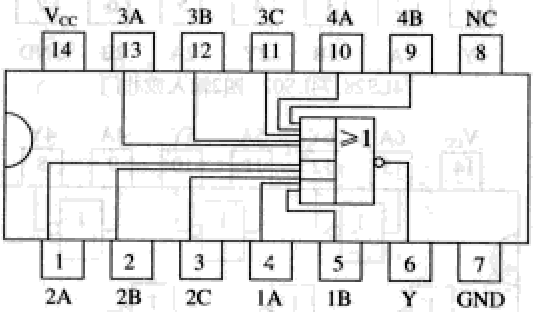
74LS86　四2输入异或门



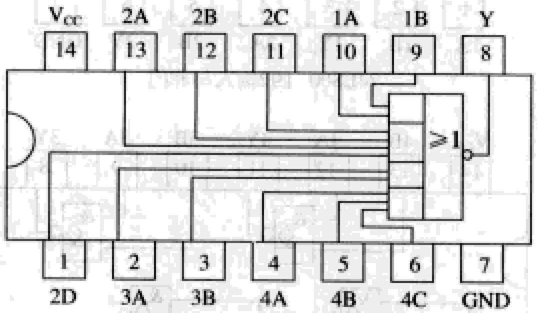
74LS30　8输入与非门



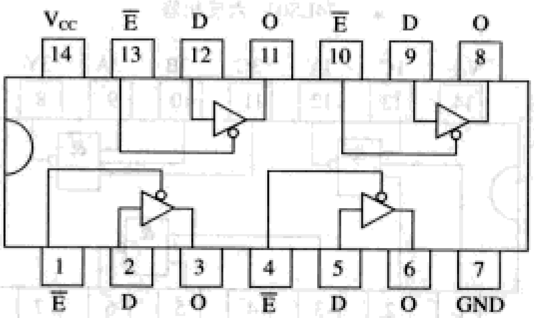
74LS32　四2输入或门



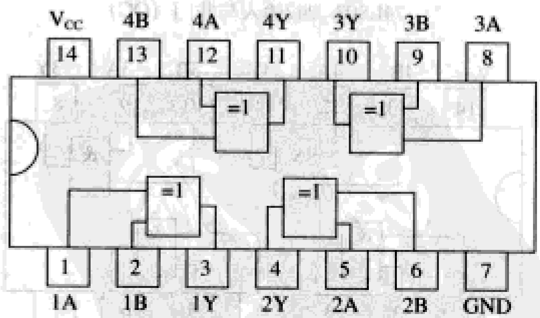
74LS54　4路2-3-3-2与或非门



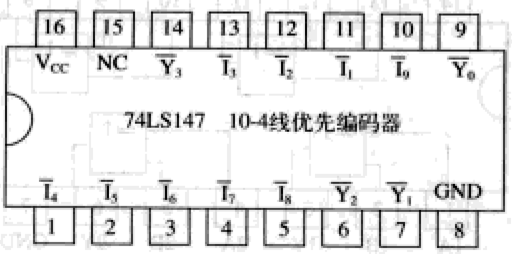
74LS64　4路2-4-2-3与或非门



74LS125　四总线缓冲器



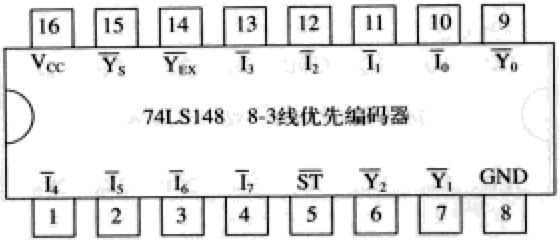
74LS386　四2输入异或门

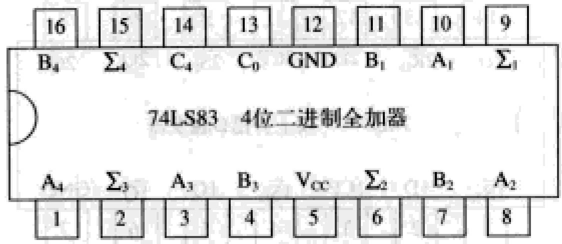


74LS147　10-4线优先编码器

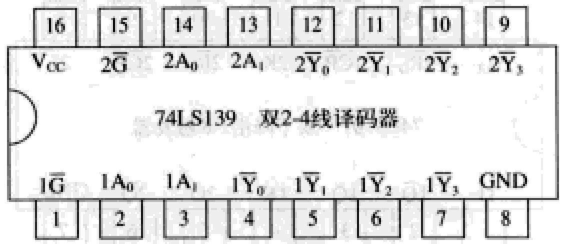


74LS48　4-7译码器/驱动器

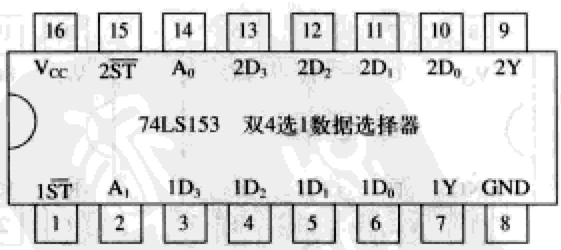

16 15 14 13 12 11 10 9
V_{CC} $\overline{Y_S}$ $\overline{Y_{EX}}$ $\overline{I_3}$ $\overline{I_2}$ $\overline{I_1}$ $\overline{I_0}$ $\overline{Y_0}$
74LS148 8-3线优先编码器
$\overline{I_4}$ $\overline{I_5}$ $\overline{I_6}$ $\overline{I_7}$ $\overline{ST}$ $\overline{Y_2}$ $\overline{Y_1}$ GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
B_4 Σ_4 C_4 C_0 GND B_1 A_1 Σ_1
74LS83 4位二进制全加器
A_4 Σ_3 A_3 B_3 V_{CC} Σ_2 B_2 A_2
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{CC} $\overline{Y_0}$ $\overline{Y_1}$ $\overline{Y_2}$ $\overline{Y_3}$ $\overline{Y_4}$ $\overline{Y_5}$ $\overline{Y_6}$
74LS138 3-8线译码器
A_0 A_1 A_2 $\overline{G_{2A}}$ $\overline{G_{2B}}$ G_1 $\overline{Y_7}$ GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{CC} $2\overline{G}$ $2A_0$ $2A_1$ $2\overline{Y_0}$ $2\overline{Y_1}$ $2\overline{Y_2}$ $2\overline{Y_3}$
74LS139 双2-4线译码器
$1\overline{G}$ $1A_0$ $1A_1$ $1\overline{Y_0}$ $1\overline{Y_1}$ $1\overline{Y_2}$ $1\overline{Y_3}$ GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{CC} D_4 D_5 D_6 D_7 A_0 A_1 A_2
74LS151 8选1数据选择器
D_3 D_2 D_1 D_0 Y $\overline{W}$ $\overline{ST}$ GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{CC} $2\overline{ST}$ A_0 $2D_3$ $2D_2$ $2D_1$ $2D_0$ 2Y
74LS153 双4选1数据选择器
$1\overline{ST}$ A_1 $1D_3$ $1D_2$ $1D_1$ $1D_0$ 1Y GND
1 2 3 4 5 6 7 8


14 13 12 11 10 9 8
1J $1\overline{Q}$ 1Q GND 2K 2Q $2\overline{Q}$
74LS73A 双JK触发器
$\overline{1CP}$ $\overline{1Rd}$ 1K V_{CC} $\overline{2CP}$ $\overline{2Rd}$ 2J
1 2 3 4 5 6 7

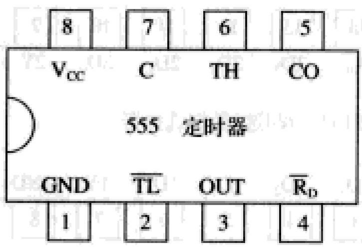

16 15 14 13 12 11 10 9
V_{CC} $\overline{1Rd}$ $\overline{2Rd}$ $\overline{2CP}$ 2K 2J $\overline{2Sd}$ 2Q
74LS112 双JK触发器
$\overline{1CP}$ 1K 1J $\overline{1Sd}$ 1Q $1\overline{Q}$ $2\overline{Q}$ GND
1 2 3 4 5 6 7 8


14 13 12 11 10 9 8
V_{CC} $2\overline{R}_D$ 2D 2CP $2\overline{S}_D$ 2Q $2\overline{Q}$
74LS74 双上升沿D触发器
$1\overline{R}_D$ 1D 1CP $1\overline{S}_D$ 1Q $1\overline{Q}$ GND
1 2 3 4 5 6 7

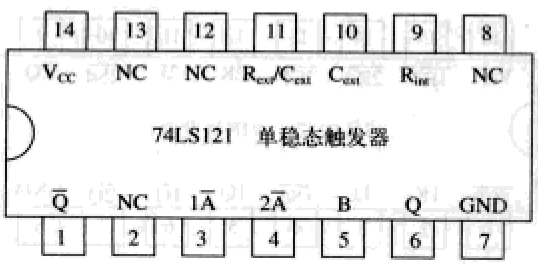

16 15 14 13 12 11 10 9
1K 1Q $1\overline{Q}$ GND 2K 2Q $2\overline{Q}$ 2J
74LS76/74LS106 双J-K触发器
1CP $1\overline{S}_D$ $1\overline{R}_D$ 1J V_{CC} 2CP $2\overline{S}_D$ $2\overline{R}_D$
1 2 3 4 5 6 7 8


14 13 12 11 10 9 8
V_{CC} $1\overline{R}_D$ $1\overline{CP}$ 2K $2\overline{R}_D$ $2\overline{CP}$ 2J
74LS107 双下降沿J-K触发器
1J $1\overline{Q}$ 1Q 1K 2Q $2\overline{Q}$ GND
1 2 3 4 5 6 7


16 15 14 13 12 11 10 9
V_{CC} A_3 B_2 A_2 A_1 B_1 A_0 B_0
74LS85 数值比较器
B_3 $I_{A<B}$ $I_{A=B}$ $I_{A>B}$ $O_{A>B}$ $O_{A=B}$ $O_{A<B}$ GND
1 2 3 4 5 6 7 8

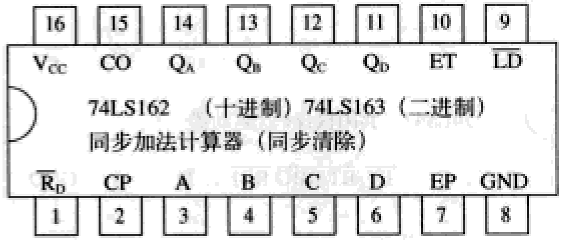

8 7 6 5
V_{CC} C TH CO
555 定时器
GND $\overline{TL}$ OUT $\overline{R}_D$
1 2 3 4


20 19 18 17 16 15 14 13 12 11
V_{CC} 8Q 8D 7D 7Q 6Q 6D 5D 5Q CP
74LS273 八D锁存器
$\overline{R}_D$ 1Q 1D 2D 2Q 3Q 3D 4D 4Q GND
1 2 3 4 5 6 7 8 9 10


14 13 12 11 10 9 8
V_{CC} NC NC R_{ext}/C_{ext} C_{ext} R_{int} NC
74LS121 单稳态触发器
$\overline{Q}$ NC $1\overline{A}$ $2\overline{A}$ B Q GND
1 2 3 4 5 6 7


14 13 12 11 10 9 8
V_{CC} 2C 2TH 2CO $2\overline{R}_D$ 2OUT $2\overline{TR}$
556 双定时器
1C 1TH 1CO $1\overline{R}_D$ 1OUT $1\overline{TR}$ GND
1 2 3 4 5 6 7

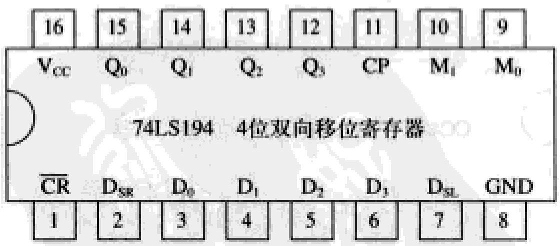

16 15 14 13 12 11 10 9
V_{CC} CO Q_0 Q_1 Q_2 Q_3 CT_T $\overline{LD}$
74LS160 （十进制）74LS161（二进制）
同步加法计算器（异步清零）
$\overline{CR}$ CP D_0 D_1 D_2 D_3 CT_P GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{CC} CO Q_A Q_B Q_C Q_D ET $\overline{LD}$
74LS162 （十进制）74LS163（二进制）
同步加法计算器（同步清除）
$\overline{R}_D$ CP A B C D EP GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{CC} D_0 CR $\overline{BO}$ $\overline{CO}$ $\overline{LD}$ D_2 D_3
74LS192 （十进制）74LS193（4位二进制）
同步加/减计数器
D_1 Q_1 Q_0 CP_D CP_U Q_2 Q_3 GND
1 2 3 4 5 6 7 8
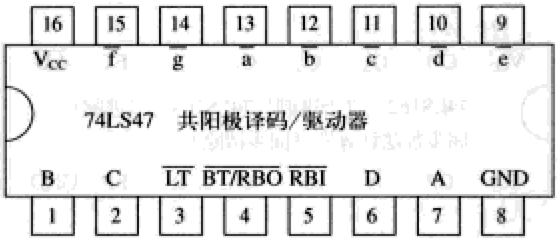

14 13 12 11 10 9 8
CP_A NC Q_0 Q_3 GND Q_1 Q_2
74LS90 十进制计数器
CP_B $R_{0(1)}$ $R_{0(2)}$ NC V_{CC} $R_{9(1)}$ $R_{9(2)}$
1 2 3 4 5 6 7


14 13 12 11 10 9 8
V_{CC} Q_0 Q_1 Q_2 Q_3 $\overline{CP}_1$ $\overline{CP}_2$
74LS95 4位移位寄存器
D_S D_0 D_1 D_2 D_3 M GND
1 2 3 4 5 6 7


16 15 14 13 12 11 10 9
V_{CC} Q_0 Q_1 Q_2 Q_3 CP M_1 M_0
74LS194 4位双向移位寄存器
$\overline{CR}$ D_{SR} D_0 D_1 D_2 D_3 D_{SL} GND
1 2 3 4 5 6 7 8

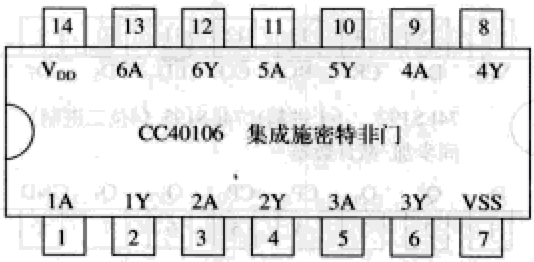

16 15 14 13 12 11 10 9
V_{CC} A_0 A_1 A_2 A_3 $\overline{9}$ $\overline{8}$ $\overline{7}$
74LS42 4-10译码器
$\overline{0}$ $\overline{1}$ $\overline{2}$ $\overline{3}$ $\overline{4}$ $\overline{5}$ $\overline{6}$ GND
1 2 3 4 5 6 7 8

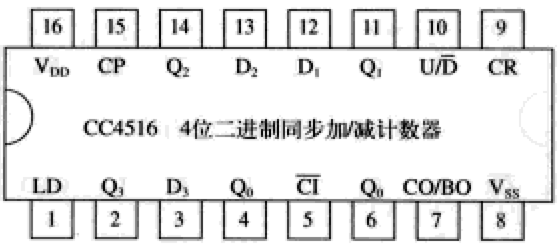

16 15 14 13 12 11 10 9
V_{CC} A_3 B_2 A_2 A_1 B_1 A_0 B_0
74LS85 数值比较器
B_3 $I_{A<B}$ $I_{A=B}$ $I_{A>B}$ $O_{A>B}$ $O_{A=B}$ $O_{A<B}$ GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
VCC f g a b c d e
74LS47 共阳极译码/驱动器
B C LT BT/RBO RBI D A GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
VCC f g a b c d e
74LS248 共阴极译码/驱动器
B C LT BT/RBO RBI D A GND
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
VDD T1 T2 CD AI BI Q Q
CC4528 双单稳态触发器
T1 T2 CD AI BI Q Q VSS
1 2 3 4 5 6 7 8


14 13 12 11 10 9 8
VDD 6A 6Y 5A 5Y 4A 4Y
CC40106 集成施密特非门
1A 1Y 2A 2Y 3A 3Y VSS
1 2 3 4 5 6 7


16 15 14 13 12 11 10 9
VDD CP Q2 D2 D1 Q1 U/D CR
CC4516 4位二进制同步加/减计数器
LD Q3 D3 Q0 CI Q0 CO/BO VSS
1 2 3 4 5 6 7 8

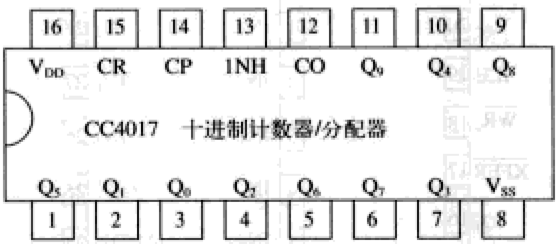

14 13 12 11 10 9 8
VDD 3A 3B 3C 3Y 1Y 1C
CC4073 三3输入与门
1A 1B 2A 2B 2C 2Y VSS
1 2 3 4 5 6 7

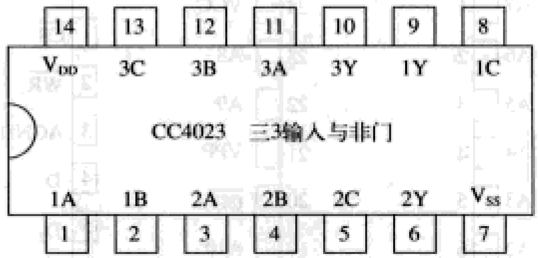

14 13 12 11 10 9 8
VDD 4A 4B 4Y 3Y 3B 3A
CC4001 四2输入或非门
1A 1B 1Y 2Y 2A 2B VSS
1 2 3 4 5 6 7


14 13 12 11 10 9 8
VDD 4B 4A 4Y 3Y 3B 3A
CC4011 四2输入与非门
1A 1B 1Y 2Y 2A 2B VSS
1 2 3 4 5 6 7


14 13 12 11 10 9 8
V_{DD} 2Q $2\bar{Q}$ 2CP $2R_D$ 2D $2S_D$
CC4013 双上升沿D触发器
1Q $1\bar{Q}$ 1CP $1R_D$ 1D $1S_D$ V_{SS}
1 2 3 4 5 6 7


16 15 14 13 12 11 10 9
V_{DD} CR CP INH CO Y_4 Y_7 NC
CC4022 八进制计数器/脉冲分配器
Y_1 Y_0 Y_2 Y_5 Y_6 NC Y_3 V_{SS}
1 2 3 4 5 6 7 8


16 15 14 13 12 11 10 9
V_{DD} CR CP INH CO Q_9 Q_4 Q_8
CC4017 十进制计数器/分配器
Q_5 Q_1 Q_0 Q_2 Q_6 Q_7 Q_3 V_{SS}
1 2 3 4 5 6 7 8

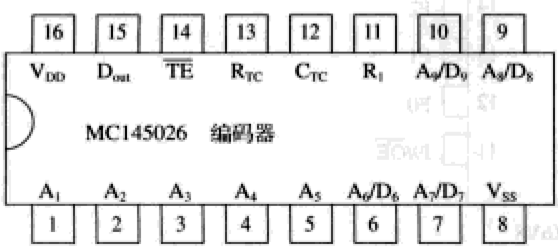

14 13 12 11 10 9 8
V_{DD} 3C 3B 3A 3Y 1Y 1C
CC4023 三3输入与非门
1A 1B 2A 2B 2C 2Y V_{SS}
1 2 3 4 5 6 7


16 15 14 13 12 11 10 9
V_{DD} 2Q $2\bar{Q}$ 2CP $2R_D$ 2K 2J $2S_D$
CC4027 双上升沿J-K触发器
1Q $1\bar{Q}$ 1CP $1R_D$ 1K 1J $1S_D$ V_{SS}
1 2 3 4 5 6 7 8

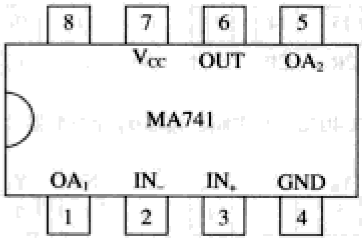

14 13 12 11 10 9 8
V_{DD} 4B 4A 4Y 3Y 3B 3A
CC4070 四异或门
1A 1B 1Y 2Y 2A 2B V_{SS}
1 2 3 4 5 6 7

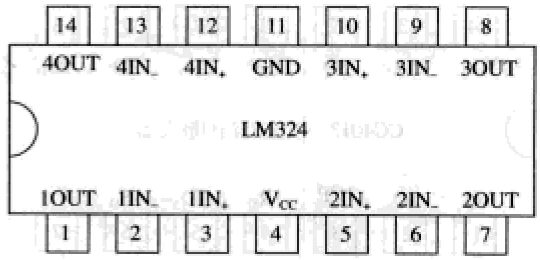

16 15 14 13 12 11 10 9
V_{DD} Y_f Y_g Y_a Y_b Y_c Y_d Y_e
CC4511 4-7段锁存译码器/驱动器
A_1 A_2 $\overline{LT}$ $\overline{BI}$ LE A_3 A_0 V_{SS}
1 2 3 4 5 6 7 8

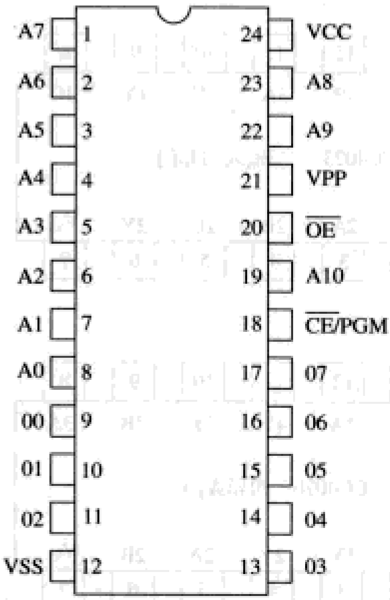

14 13 12 11 10 9 8
V_{DD} 6A 6Y 5A 5Y 4A 4Y
CC4069 六反相器
1A 1Y 2A 2Y 3A 3Y V_{SS}
1 2 3 4 5 6 7


16 15 14 13 12 11 10 9
V_{DD} D_{out} $\overline{TE}$ R_{TC} C_{TC} R_1 A_9/D_9 A_8/D_8
MC145026 编码器
A_1 A_2 A_3 A_4 A_5 A_6/D_6 A_7/D_7 V_{SS}
1 2 3 4 5 6 7 8

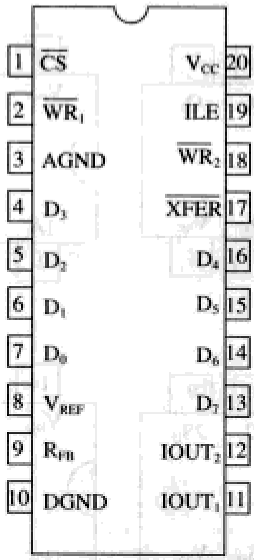

16 15 14 13 12 11 10 9
V_{DD} D_6 D_7 D_8 D_9 V_T R_2/C_2 D_{in}
MC145027 译码器
A_1 A_2 A_3 A_4 A_5 R_1 C_1 V_{SS}
1 2 3 4 5 6 7 8







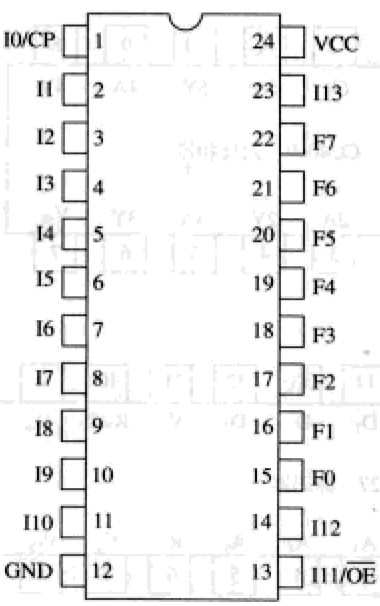
EPROM2716



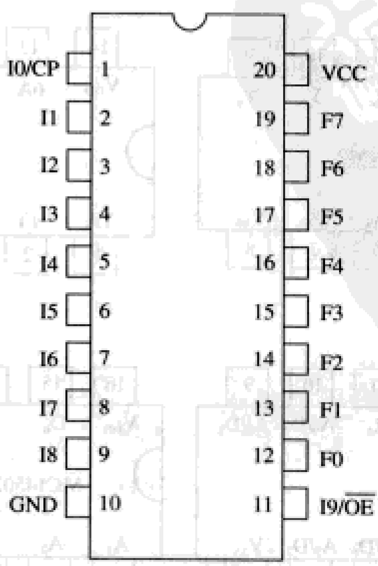
DAC0832 D/A转换器



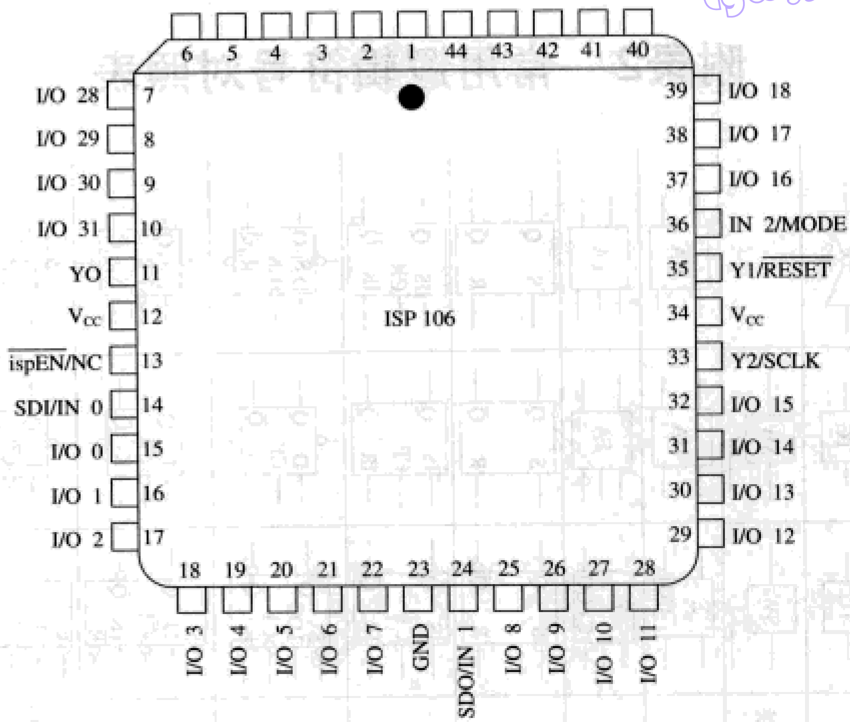
ADC0808/0809 A/D转换器

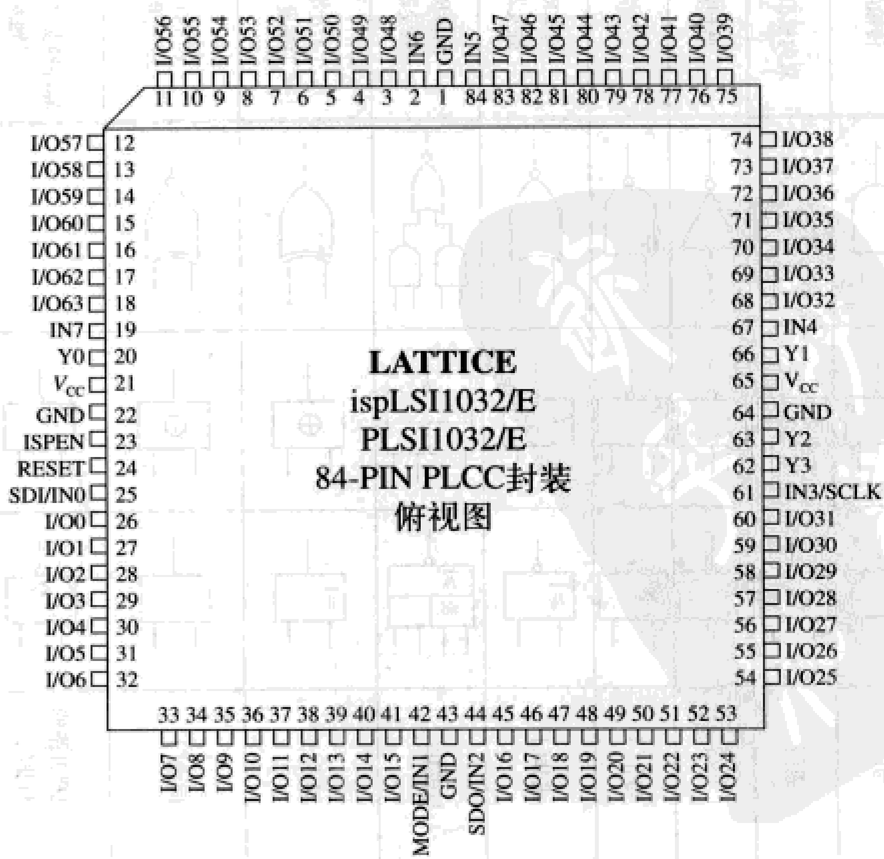


GAL20V8



GAL16V8


ISP 106
I/O 28 7
I/O 29 8
I/O 30 9
I/O 31 10
YO 11
Vcc 12
ispEN/NC 13
SDI/IN 0 14
I/O 0 15
I/O 1 16
I/O 2 17
I/O 3 18
I/O 4 19
I/O 5 20
I/O 6 21
I/O 7 22
GND 23
SDO/IN 1 24
I/O 8 25
I/O 9 26
I/O 10 27
I/O 11 28
I/O 12 29
I/O 13 30
I/O 14 31
I/O 15 32
Y2/SCLK 33
Vcc 34
Y1/RESET 35
IN 2/MODE 36
I/O 16 37
I/O 17 38
I/O 18 39
6 5 4 3 2 1 44 43 42 41 40


LATTICE
ispLSI1032/E
PLSI1032/E
84-PIN PLCC封装
俯视图
I/O56 11
I/O55 10
I/O54 9
I/O53 8
I/O52 7
I/O51 6
I/O50 5
I/O49 4
I/O48 3
IN6 2
GND 1
IN5 84
I/O47 83
I/O46 82
I/O45 81
I/O44 80
I/O43 79
I/O42 78
I/O41 77
I/O40 76
I/O39 75
I/O57 12
I/O58 13
I/O59 14
I/O60 15
I/O61 16
I/O62 17
I/O63 18
IN7 19
Y0 20
Vcc 21
GND 22
ISPEN 23
RESET 24
SDI/IN0 25
I/O0 26
I/O1 27
I/O2 28
I/O3 29
I/O4 30
I/O5 31
I/O6 32
I/O7 33
I/O8 34
I/O9 35
I/O10 36
I/O11 37
I/O12 38
I/O13 39
I/O14 40
I/O15 41
MODE/IN1 42
GND 43
SDO/IN2 44
I/O16 45
I/O17 46
I/O18 47
I/O19 48
I/O20 49
I/O21 50
I/O22 51
I/O23 52
I/O24 53
I/O25 54
I/O26 55
I/O27 56
I/O28 57
I/O29 58
I/O30 59
I/O31 60
IN3/SCLK 61
Y3 62
Y2 63
GND 64
Vcc 65
Y1 66
IN4 67
I/O32 68
I/O33 69
I/O34 70
I/O35 71
I/O36 72
I/O37 73
I/O38 74

附录2 常用逻辑符号对照表

常用逻辑符号对照表

名称	国际符号	曾用符号	国外流行符号
与门	&		
或门	≥1	+	
非门	1		
与非门	&		
或非门	≥1	+	
与或非门	& ≥1	+	
异或门	=1	⊕	
同或门	=	⊙	
集电极开路的与非门	&		
三态输出的非门	1 ▽ EN		

名称	国际符号	曾用符号	国外流行符号
传输门	TG	TG	
双向模拟开关	SW	SW	
半加器	Σ CO	HA	HA
全加器	Σ CI CO	FA	FA
基本RS触发器	S Q R $\bar{Q}$	S Q R $\bar{Q}$	S Q R $\bar{Q}$
同步RS触发器	1S C1 1R Q $\bar{Q}$	1S CP 1R Q $\bar{Q}$	1S CK 1R Q $\bar{Q}$
（上升沿）D触发器	S 1D C1 R Q $\bar{Q}$	D CP Q $\bar{Q}$	D $\bar{S}_D$ CK $\bar{R}_D$ Q $\bar{Q}$
（下降沿）JK触发器	S 1J C1 1K R Q $\bar{Q}$	J CP K Q $\bar{Q}$	J $\bar{S}_D$ CK K $\bar{R}_D$ Q $\bar{Q}$
脉冲触发（主从）JK触发器	S 1J C1 1K R Q $\bar{Q}$	J CP K Q $\bar{Q}$	J $\bar{S}_D$ CK K $\bar{R}_D$ Q $\bar{Q}$
带施密特触发特性的与门	&⎍	⎍	⎍

参 考 文 献

1. 白中英等，数字逻辑与数字系统题解、题库与实验，北京：科学出版社，2002
2. 陈汝全主编，电子技术常用器件应用手册，北京：机械工业出版社，2002
3. 黄正谨等，CPLD系统设计技术入门与应用，北京：电子工业出版社，2002
4. 刘润华等，电子设计自动化EDA，东营：石油大学出版社，2001
5. 卢结成等，电子电路实验及应用课题设计，合肥：中国科学技术大学出版社，2002
6. 吕思忠等，数字电路实验与课程设计，哈尔滨：哈尔滨工程大学出版社，2001
7. 毛期俭主编，数字逻辑与数字系统实验及应用，北京：人民邮电出版社，2005
8. 潘松，黄继业编著，EDA技术实用教程，北京：科学出版社，2002
9. 潘松，王国栋编著，VHDL实用教程（第二版），成都：电子科技大学出版社，2001
10. 田良等，综合电子设计与实践，南京：东南大学出版社，2002
11. 吴新开等，现代数字系统实践教程，北京：人民邮电出版社，2004
12. 谢自美主编，电子线路设计·实验·测试（第二版），武汉：华中理工大学出版社，2000
13. 谢自美主编，电子线路综合设计，武汉：华中科技大学出版社，2006
14. 徐志军等，CPLD/FPGA的开发与应用，北京：电子工业出版社，2002
15. 许小军主编，电子技术实验与课程设计指导—数字电路分册，南京：东南大学出版社，2005
16. 曾繁泰等，可编程器件应用导论，北京：清华大学出版社，2001
17. 赵丽红等，数字逻辑与数字系统习题解答与实验指导，北京：电子工业出版社，2005

数字电路和数字系统 实验与课程设计实训教程

本书根据教育部高等学校电子信息与电气学科教学指导委员会“数字电路与逻辑设计”课程教学基本要求编写，反映了作者丰富的实践教学和教材编写经验。取材自合肥工业大学使用多年、反应良好的院级平台课程“数字电路与数字系统实践”的讲义，并特别注意体现技术领域的新趋势和新动向。书中包含37项数字电路与系统的实验，41个课程设计综合选题，非常贴近工程实际。该讲义近年来用于本校全国大学生电子设计竞赛培训，也收到了很好的效果。

本书适用于电子信息与电气各学科和计算机等专业教学，特别适合用于单独设置的实验课程。

杨　萍　女，合肥工业大学计算机与信息学院副教授。从事电子线路课程教学与研究二十多年，华东地区高等学校工科电子线路课程教学研究会理事，兼任实验室主任和学院实验中心副主任多年。

张旭东　男，合肥工业大学计算机与信息学院副教授，中国科学技术大学博士。主要研究方向为图像处理、模式识别和智能信息处理。作为主要成员，先后参加了国家自然科学基金项目4项，主持省级科技攻关项目1项。结合项目在国内外重要学术会议和期刊发表学术论文近20余篇，其中多篇被EI收录。获科技成果奖1项，安徽省自然科学三等奖1项，申请实用新型专利1项。

罗　珣　女，合肥工业大学计算机与信息学院讲师，硕士。主要研究方向为计算机控制。讲授《数字逻辑》课程，曾在合肥工业大学青年教师基本功大赛中获得优胜奖。在“第四届全国高等学校计算机课件评比”中获优胜奖。参编多部教材。

许良凤　女，合肥工业大学计算机与信息学院讲师，硕士，从事数字电路及数字电路实验的教学多年。主要研究方向为多媒体通信和图像处理。参编2部教材。参加省级教改项目1项。

本书相关信息请访问：图灵网站 http://www.turingbook.com
读者/作者热线：(010) 88593802
反馈/投稿/推荐信箱：contact@turingbook.com

分类建议：电子电气／数字电路
人民邮电出版社网址 www.ptpress.com.cn

ISBN 978-7-115-20460-8
9 787115 204608 >
ISBN 978-7-115-20460-8/TN
定价：25.00 元